Active Learning with Effective Scoring Functions for Semi-Supervised Temporal Action Localization

Ding Li, Xuebing Yang, Yongqiang Tang, Chenyang Zhang, and Wensheng Zhang

***Abstract*—Temporal Action Localization (TAL) aims to predict both action category and temporal boundary of action instances in untrimmed videos, *i.e.*, start and end time. Fully-supervised solutions are usually adopted in most existing works, and proven to be effective. One of the practical bottlenecks in these solutions is the large amount of labeled training data required. To reduce expensive human label cost, this paper focuses on a rarely investigated yet practical task named semi-supervised TAL and proposes an effective active learning method, named AL-STAL. We leverage four steps for actively selecting video samples with high informativeness and training the localization model, named *Train, Query, Annotate, Append*. Two scoring functions that consider the uncertainty of localization model are equipped in AL-STAL, thus facilitating the video sample rank and selection. One takes entropy of predicted label distribution as measure of uncertainty, named Temporal Proposal Entropy (TPE). And the other introduces a new metric based on mutual information between adjacent action proposals and evaluates the informativeness of video samples, named Temporal Context Inconsistency (TCI). To validate the effectiveness of proposed method, we conduct extensive experiments on two benchmark datasets THUMOS'14 and ActivityNet 1.3. Experiment results show that AL-STAL outperforms the existing competitors and achieves satisfying performance compared with fully-supervised learning.**

***Index Terms*—Temporal Action Localization, Semi-supervised Learning, Active Learning, Scoring Function**

I. Introduction

With the tremendous increase of video-capture devices and video-sharing platforms, a large collection of videos have been generated, stored and shared in our daily lives. Among the video contents, the long untrimmed videos are commonly available in the wild, and there is an urgent need to intelligently parse these video data in many real-world applications, such as entertainment, visual surveillance, and robotics [1], [2]. This encourages the study of Temporal Action Localization (TAL) which requires not only classifying category of action instances, but also localizing temporal boundaries of them, *i.e.*, their start and end time.

This work was supported in part by the National Key R&D Program of China (No. 2020AAA0109600) and in part by the National Natural Science Foundation of China (No. 62106266 and No. 62006139). *(Corresponding authors: Yongqiang Tang and Wensheng Zhang).*

Ding Li and Wensheng Zhang are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China, and also with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: liding2019@ia.ac.cn; zhangwenshengia@hotmail.com)

Xuebing Yang, Yongqiang Tang and Chenyang Zhang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: {yangxuebing2013, yongqiang.tang, zhangchenyang2016}@ia.ac.cn).

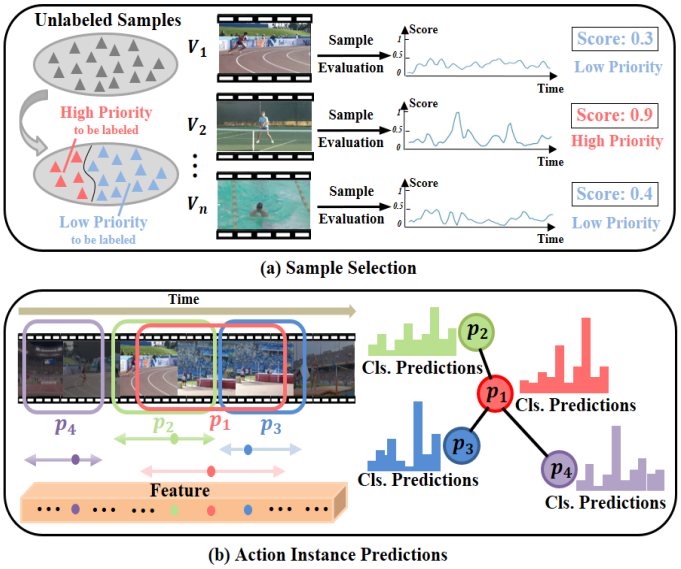


Fig. 1. Conventional semi-supervised methods overlook the discrepancy in informativeness between different video samples. (a) In AL-STAL, we assign each unlabeled sample a video-level score as the measurement of informativeness, and this score is aggregated from frame-level predictions. When labeling video data, those samples with larger score will have high priority, while the other samples are given low priority. (b) An action instance proposal is generated in each temporal location, and the relations among these proposals (red, green, blue, purple at different proposals for clarity) are conducive to informativeness estimation.

Recent years have witnessed great success in deep TAL [3]–[14], nevertheless deep localization models are hungry for a big amount of labeled data. Unfortunately, for action instances in long untrimmed videos, the precise annotations of temporal boundaries and categories require immense time and manual effort. Therefore, localization models with less supervision [15]–[23], *e.g.*, semi-supervised temporal action localization (STAL) have attracted considerable research interests recently. For instance, KFC [21] utilizes only 40% labeled data for training, and 60% annotations are employed in SSTAP [22].

Notably, conventional semi-supervised methods mainly focus on the exploration of unlabeled data, while the labeled data are usually randomly selected. With respect to random sampling, however, the contribution of labeling different samples to improving localization performance are unequal, indicating some label budgets are wasted on those samples with lower informativeness. Thus, active learning can be utilized to evaluate the informativeness of samples and prioritize labeling the informative ones other than the normal ones. Fig. 1(a) briefly depicts the process of selecting highly-informative samples.

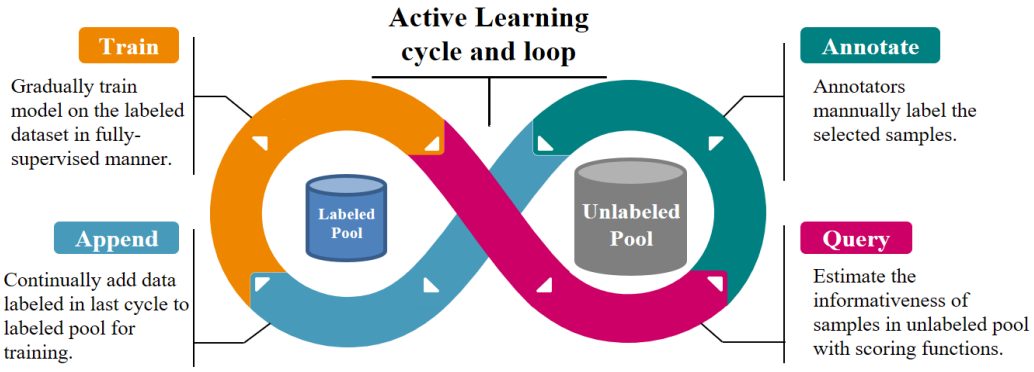


Fig. 2. Active learning procedure diagram. At each cycle, the most informative samples are selected and annotated by the scoring function in the *Query* step, and then training dataset is updated with newly annotated samples. As the labeled training set updates in the loop, the trained model evolves to have better performance with fewer label cost.

A typical active learning procedure is shown in Fig. 2, where four steps named *Train, Query, Annotate, Append* are involved in each active learning cycle. It is worth noting that the scoring function in *Query* step enables the base model itself to distinguish the informative samples from the ordinary ones actively rather than with the help of intense human efforts. As the training dataset updates with model involved, better performance can be achieved with fewer samples. When working with a continuous video stream, the active learning method is implemented as iterations of a Mobius strip. Although there have been emerging efforts to develop active learning for image classification and other visual tasks [24]–[29], there still lacks an active learning method specified for temporal action localization (active temporal action localization). In particular, there are two crucial challenges shown in Fig. 1(b). First, different from *image-level* or *video-level* predictions in previous tasks with active learning, the output of STAL are *instance-level* predictions of video segments. This motivates us to focus on measuring the informativeness of action instances first, rather than directly considering the whole video. Second, the relationship among predictions within temporal context lacks emphasis, and the neglected relationship would benefit the informativeness measurement for sample selection. Hence, naively adopting conventional active learning methods would be infeasible. How to effectively query unlabeled samples based on intrinsic characteristics of action localization remains in-depth study.

To address the issues above, we propose to develop a novel active learning method for STAL, *i.e.*, AL-STAL, to alleviate the waste of label cost by exploring the potential of the model itself. The key idea of our method is to design reasonable and effective scoring functions for sample selection based on the predictions of base localization network. Specifically, in *Query* step of active learning procedure, we propose two scoring functions that considers the uncertainty of localization model.

- *Temporal Proposal Entropy (TPE).* This function focuses on the temporal proposals (action instances) generated in each time step, and takes the entropy of predicted label distribution as measure of uncertainty. High entropy indicates the classifier is uncertain about the category of this temporal proposal.
- *Temporal Context Inconsistency (TCI).* This function introduces a frame-level metric which considers the mutual information between the adjacent action proposals. The hypothesis of this function is that divergence between predicted probability distributions will be large if predictions are locally inconsistent within the temporal context, and vice versa.

In this way, only a small number of informative samples are labeled, and the localization performance could be increasingly improved. The main contribution of this paper can be summarized as follows:

1) We propose an active learning algorithm specified for semi-supervised temporal action localization, which employs the knowledge of network to select samples for annotating. Compared with passive learning with random sampling, our algorithm can leverage more informative labeled samples, and achieve better performance when given the fixed label budget.
2) We present two scoring functions (TPE and TCI) with different quantitative metrics, which facilitate to evaluate the informativeness of unlabeled video samples. By introducing proposal entropy and context inconsistency along the temporal axis, AL-TAL allows us to automatically mine samples which benefits the training of temporal action localizer.
3) We evaluate our proposed method on popular benchmark datasets for temporal action localization, *e.g.* THUMOS'14 and ActivityNet 1.3, and both metrics for action localization and label cost saving are utilized for evaluation. Experiment results verify the effectiveness of our proposed method.

The rest of this paper is organized as follows. In Section II, we briefly review some related works, followed by the elaboration of proposed AL-STAL method in Section III. Experiment results and analyses are shown in Section IV. Finally, we conclude this paper in Section V.

II. Related Works

In order to indicate our proposed method, we will review two related areas in this section. We will first review the task of

temporal action localization. Then, related advances in active learning methods will be reviewed.

A. Temporal Action Localization

Temporal action localization aims at localizing boundaries and classifying category of action instances in untrimmed videos. Previous methods can be divided into two categories: one-stage and two-stage. The one-stage methods integrate proposal generation and classification into an end-to-end structure, and achieve higher efficiency [30], [31]. TURN [32] aggregated features from basic video unit for clip-level features, which are used to classify the activity and regress the temporal boundary. R-C3D [8] took the inspiration from faster-RCNN [33] and utilized a streamline including proposal generation, proposal-wise pooling and final prediction. GTAN [9] modified the pooling procedure, adopting a weighted average via a learnable gaussian kernel for each proposal. AFSD [7] proposed a purely anchor-free model with saliency-based refinement module and boundary pooling mechanism, and obtained boundary-sensitive features for action localization. By contract, the two-stage method first generates action instances with temporal boundaries and followed by classifier. These works mainly focus on evaluating 'actionness', which indicates the probability of a potential action, for each frame or clip in a video. BSN [3] adopted three activeness curves to locate flexible proposal boundaries, and a Boundary-Matching confidence map in BMN [4] was introduced to generate better proposals. PGCN [5] used graph network to extract features between different proposals, while DBG [6] used two feature maps separately for completeness regression and temporal boundary classification. Nevertheless, a large amount of labeled data are required for model training, such methods are thus not economical when applied in practical scenarios.

To alleviate the problem of large label cost, a few annotations accompanied with a large number of unlabeled videos are adopted for semi-supervised training in recent works [21]–[23]. KFC [21] added perturbations to each feature along temporal axis and employed consistency regularization to encourage the model to retain this observation. Besides, some works intend to generate action proposals, and then performs localization. For example, Ji $et\ al.$ [23] is the first to incorporate semi-supervised learning in action proposals generation to achieve the label efficiency. SSTAP [22] designed a temporal-aware semi-supervised branch and a relation-aware self-supervised branch to better explore unlabeled videos. Different from randomly labeling a part of samples in previous semi-supervised learning methods, our AL-STAL employs the knowledge of localizer itself and prioritize labeling the informative samples other than the normal ones.

B. Active Learning

Active learning aims to select the most useful samples from the unlabeled dataset and hand it over to the annotator for labeling, so as to reduce the cost of labeling as much as possible while still maintaining performance [34]. The key hypothesis is that, if the learning algorithm is allowed to choose the data from which it learns to be "curious", it will perform better with less training data [35]. Active learning methods have been used in many areas, such as speech recognition, information extraction and computer vision [36]. Here, we mainly focus on active learning in computer vision.

Most previous works are focused on image classification, such as [24], [26], [27], [28], [37], [38] and [39]. For example, Gal [38] proposed an active learning method that obtains uncertainty estimation through multiple forward passes with Monte Carlo Dropout. Yoo [39] created a module that learns to predict loss of input images and selected the unlabeled image with the higher predicted loss. CEAL [24] assigned pseudo-labels to samples with high confidence and added them to the highly uncertain sample set queried using the uncertainty-based active learning method, then used the expanded training set to train the active image classifier. Also, there are very few works of deep active learning for object detection [29], [40], [41]. [41] considered both the uncertainty and the robustness of the object detector, ensuring that the network performs accurately in all classes. [29] constructed a novel active learning method based on the "query by committee" paradigm which improves the performance with less label cost. Compared with images, which require only the processing of spatial features, video tasks need to process temporal features. This makes the work of annotating video tasks more expensive, driving introducing active learning more urgent. DeActive [42] proposed an active activity recognition model and accumulated the most informative and meaningful labeled samples. Compared with the traditional activity recognition model, DeActive requires fewer labeled samples, consumes less resources, and achieves high recognition accuracy. USAP [43] proposed a novel uncertainty sampling algorithm for action recognition using expected Average Precision (AP), and calculated the expected AP using dynamic programming in polynomial time. Despite of the existing progress in these research areas described above, active learning for TAL is rarely studied. When facing with temporally-correlated action instance, directly adopting previous active learning method to train an action localizer would achieve limited performance.

III. Methods

In this section, we will formally introduce AL-STAL. To facilitate understanding, Section III-A first provides necessary notations and overview of AL-STAL. Then, Section III-B introduces the base model for TAL. Next, Section III-C depicts a frame-level scoring function for sample selection, named Temporal Proposal Entropy (TPE). And Section III-D depicts the other frame-level scoring function, named Temporal Context Inconsistency (TCI). Finally, Section III-E describes how to aggregate frame-level score into video-level score.

A. Preliminary and Overview

Initially, a video action dataset χ is divided into a small set of labeled video samples χ_L^0 and a large set of unlabeled video samples χ_U^0, where each sample (V_i, Ψ_{V_i}) contains a video $V_i = \left\{v_t^i\right\}_{t=1}^T$ with T RGB frames or optical flows and the corresponding annotation Ψ_{V_i}. Ψ_{V_i} can be depicted as tuples $\Psi_{V_i} = \left\{\left(\tau_{n,i}^s, \tau_{n,i}^e, y_n^i\right)\right\}_{n=1}^{N_i}$, where N_i is the number

of action instances in V_i, $\tau^s_{n,i}, \tau^e_{n,i}$ denote the start and end time of the n-th action instance respectively, $y^i_n \in \mathbb{R}^C$ is the action category, C is the number of category. Thus, $\chi^0_L = \{(V_i, \Psi_{V_i})\}_{i=1}^{N_L}$, $\chi^0_U = \{V_i\}_{i=1}^{N_U}$, where N_L and N_U are the corresponding number of videos in labeled and unlabeled sets.

Assume that there are M active learning cycles in model training, different amounts of data are labeled in each cycle. Then a set of localization networks are trained using corresponding labeled data in these cycles, and can be defined as $\{\boldsymbol{\Phi_1}, \boldsymbol{\Phi_2}, \cdots, \boldsymbol{\Phi_M}\}$. Also, during the training process of M cycles, mining a subset of samples from the unlabeled data pool χ_U and appending them to the labeled set χ_L after annotating are included in each cycle. For instance, a localization network $\boldsymbol{\Phi_0}$ is firstly trained on initial labeled set χ^0_L. With trained model Φ_0 in the initial cycle, active learning aims to select a set of videos χ^0_A from χ^0_U to be manually annotated and append them to the updated labeled set χ^1_L, depicted as $\chi^1_L = \chi^0_L \cup \chi^0_A$. To obtain the appropriate χ^0_A, a video-level score s is introduced for evaluating the informativeness of unlabeled video samples. Using our defined score, unlabeled videos are sorted in descending order for sample selection, and χ^0_A consists of the k videos with higher scores. Next, a new localization model $\boldsymbol{\Phi_1}$ is trained with updated labeled set χ^1_L. The localization model training and sample selection process are repeated for M cycles, and finally a well-trained localization model $\boldsymbol{\Phi_M}$ is obtained. The pipeline of M cycles is shown in Fig. 3.

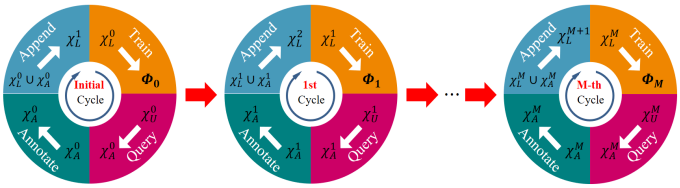


Fig. 3. The iteration of M active learning cycles. In these cycles, samples in χ_A are annotated, and unlabeled pool χ_U and labeled pool χ_L are updated.

The selected video samples are expected to be the most informative ones for model training and are able to promote the localization performance as much as possible. Due to the limited annotation budget, usually only a small number of informative videos are selected, which raises a key issue: *how to measure the informativeness of video samples.* To tackle with this issue, two scoring functions are proposed to evaluate and select highly-informative videos based on localization network output, which will be defined in Section III-C and Section III-D. Below, we give the details of AL-STAL. Fig. 5 shows a brief description of the procedure in the m-th cycle, which will be elaborated subsequently. And Algorithm 1 demonstrates the algorithmic steps of our proposed method.

B. Action Localization Network

Network Architecture. As for base model for TAL, we construct an I3D-based network with classification head and regression head like AFSD [7], a brief introduction is shown in Fig. 4. Specifically, the base model takes a video V_i as input, and mainly composes of three parts: feature extraction,

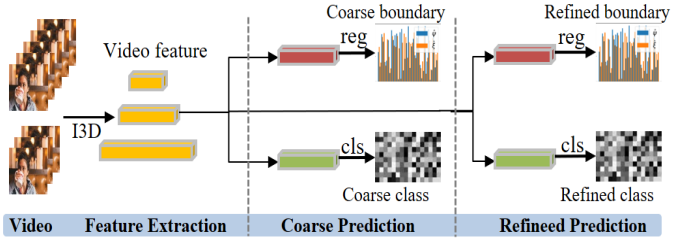


Fig. 4. Base model for TAL. Taking a video V_i as input, we extract temporal pyramid features with I3D model. Next, features are utilized to generate coarse proposals via basic prediction module. Finally, the class scores, start and end boundaries of coarse proposals are adjusted by refinement module.

coarse prediction and refined prediction. Firstly, a Kinetics pre-trained I3D [44] model is adopted to extract 3D video feature $F_i \in \mathbb{R}^{T\times D\times H\times W}$, where T, D, H, W denote the time step, feature dimension, height and width respectively. This feature is afterwards flattened along the last three dimensions to a 1D feature sequence which contains the temporal and spatial information of whole video. A feature pyramid network (FPN) is introduced to extract features in different length of temporal scales. Then, the extracted pyramid features are further utilized to generate a coarse proposal sequence with a basic anchor-free prediction module at each time step, which includes a simple regressor and classifier. At last, a fine-grained prediction for both temporal regression and action classification is output based on the extracted video feature.

For instance, the feature of the l-th FPN level is denoted as $f_{i,l} \in \mathbb{R}^{T_l\times D}$, where T_l represents the length of sequence in this level. $f_{i,l}$ is embedded into two latent space and processed with one layer of temporal convolution to get coarse start and end boundary distances $\left(\hat{d}^s_{i,l,t}, \hat{d}^e_{i,l,t}\right)$ and coarse class score $\hat{y}^{coarse}_{i,l,t}$ for each location t. The start and end time for t-th time step in l-th level are as follow:

$$\begin{aligned}\hat{\tau}^s_{i,l,t} &= t\times 2^l - \hat{d}^s_{i,l,t},\\ \hat{\tau}^e_{i,l,t} &= t\times 2^l - \hat{d}^e_{i,l,t}\,.\end{aligned} \tag{1}$$

Through the simple network in anchor-free manner, T_l proposals are generated for l-th FPN layer in all, coarse predictions of action category $\hat{y}^{coarse}_{i,l,t}$ and temporal boundary $\left(\hat{\tau}^s_{i,l,t}, \hat{\tau}^e_{i,l,t}\right)$ are output. Next, a saliency-based refinement module Θ is designed for further improve video proposal classification and boundary regression. The refinement module predict offsets of regression $\left(\Delta\hat{\tau}^s_{i,l,t}, \Delta\hat{\tau}^e_{i,l,t}\right)$, which can be added to the coarse predictions to get a fine-grained ones $\left(\tilde{\tau}^s_{i,l,t}, \tilde{\tau}^e_{i,l,t}\right)$, and refined class score $\hat{y}^{refine}_{i,l,t}$.

Training. The localization model can be optimized with the following objective function:

$$Loss = l^{coarse}_{cls} + \lambda\, l^{coarse}_{loc} + l^{refine}_{cls} + l^{refine}_{loc} + \gamma l_q, \tag{2}$$

where λ, γ are hyper-parameters. The generated proposal at t-th time step corresponds to the ground truth segment $\left(\tau^s_{i,l,t}, \tau^e_{i,l,t}, y_{i,l,t}\right)$, then l^c_{cls}, l^r_{cls} are softmax focal loss [45] between both classification prediction $\left\{\hat{y}^c_{i,l,t}, \hat{y}^r_{i,l,t}\right\}$ and ground truth labels $y_{i,l,t}$. l^{coarse}_{loc} is a tIoU loss between coarse boundaries and ground truth, l^{refine}_{loc} is a L1 loss between

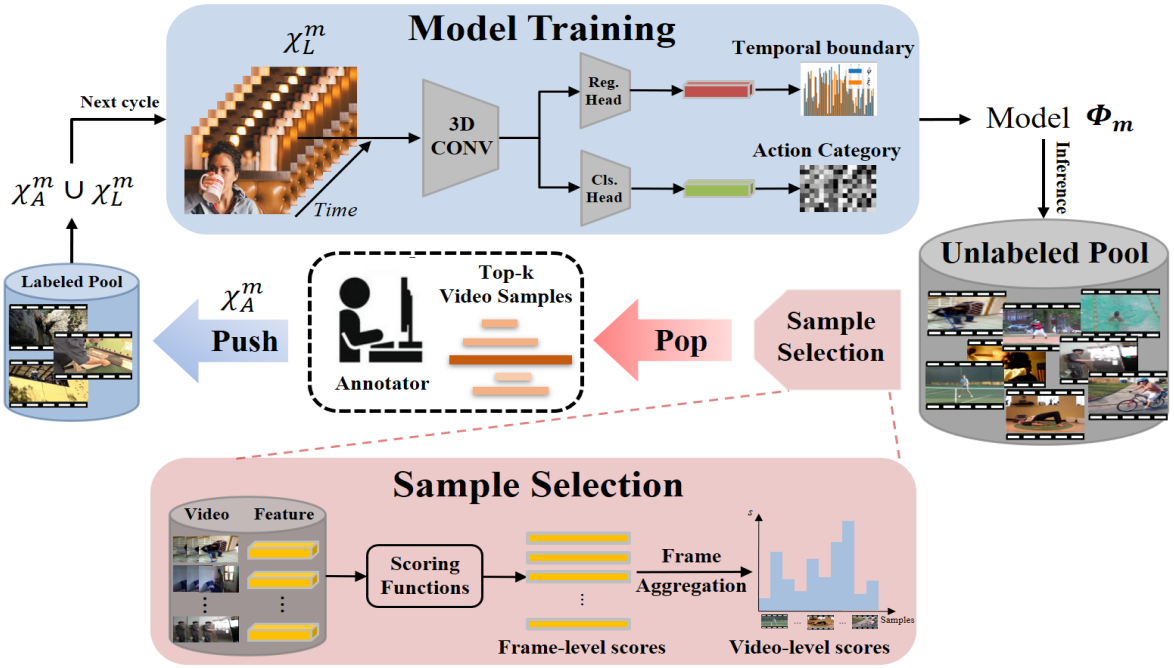


Fig. 5. The Framework of our Active Learning for Semi-supervised Temporal Action Localization (AL-STAL). We first train the TAL model with initial labeled data. Sample selection is then accomplished in support of popping videos from the unlabeled pool, where scoring functions are designed for estimating the probability of samples be selected. Finally, the selected samples are pushed in the labeled pool for further training. The red and blue arrows indicate the *Query* step and *Append* step in active learning loop respectively.

predicted offset and the corresponding offset label. l_q is a quality loss used to suppress the proposals with low quality.

Inference. For the t-th temporal location in l-th FPN layer, the final predictions are formalized through all outputs from our model, including coarse predictions $\hat{\tau}^s_{i,l,t}, \hat{\tau}^e_{i,l,t}, \hat{y}^c_{i,l,t}$ and refined ones $\Delta\hat{\tau}^s_{i,l,t}, \Delta\hat{\tau}^e_{i,l,t}, \hat{y}^r_{i,l,t}$, in the following form:

$$\begin{aligned}
\hat{\omega}_{i,l,t} &= \tilde{\tau}^e_{i,l,t} - \tilde{\tau}^s_{i,l,t}, \\
\tilde{\tau}^e_{i,l,t} &= \hat{\tau}^e_{i,l,t} + \frac{1}{2}\,\hat{\omega}_{i,l,t}\,\Delta\hat{\tau}^e_{i,l,t}, \\
\tilde{\tau}^s_{i,l,t} &= \hat{\tau}^s_{i,l,t} + \frac{1}{2}\,\hat{\omega}_{i,l,t}\,\Delta\hat{\tau}^s_{i,l,t}, \\
\tilde{y}_{i,l,t} &= \frac{1}{2}\left(\hat{y}^c_{i,l,t} + \hat{y}^r_{i,l,t}\right).
\end{aligned} \tag{3}$$

Soft-NMS [46] is used to assemble all predictions and suppress redundant proposals.

C. Temporal Proposal Entropy (TPE) for Sample Selection

Entropy is an information-theoretic measure which represents the amount of information needed to "encode" a probability distribution, thus, it is often considered as a reasonable measure of uncertainty. The entropy function is defined as $\mathbb{H}(p) = -\sum p_i \log(p_i)$, where p represents a probability distribution. Entropy-based active learning methods consider entropy value of the posterior probability in classification as a measure of sample uncertainty, and have been commonly used for classification problem. An intuitive way to adopt these methods for TAL is to choose the set of videos that are hard for video classification. However, in such way, the nature of action localization would be neglected. For one thing, the input of TAL is a long and untrimmed video, the expected action instances with short durations are included in the video, indicating that the difficulties between classifying the whole video and classifying the instance-level action categories would be different. For another, the expected action instance may easily be temporally localized even if the whole video tends to be misclassified. The expected action instance may easily stand out in the whole video. For example, an temporal action localizer may have no trouble in localizing an instance of rip-jump in a long video of human walking, while an action classifier may misunderstand according to the instance-level label "Jump". Therefore, we consider a set of temporal proposals for each video sample and obtain predictions corresponding to these proposals, then introduce entropy measure for sample selection.

For each video sample V_i in the unlabeled pool χ^0_L, the temporal proposal with respect to the t-th temporal location belongs to a certain class $y_{i,l,t} \in \{1, \cdots, C\}$, we can consider the class membership variable to be a *random variable* denoted by Y. According to the localization network in Section III-A, we are able to obtain a predicted posterior distribution $\tilde{y}_{i,l,t}$ for Y, and the discrete entropy of Y can be estimated by

$$\mathbb{H}(Y) = -\sum_{c=1}^{C} \tilde{y}^c_{i,l,t} \log\left(\tilde{y}^c_{i,l,t}\right). \tag{4}$$

In TPE, the estimated discrete entropy of temporal proposal is taken as the frame-level score for sample selection, defined as $s_i[t] = \mathbb{H}(Y)$, then score aggregation function is utilized to obtain the video-level score s_i.

D. Temporal Context Inconsistency (TCI) for Sample Selection

Given a long, untrimmed video V_i, we firstly choose a video segment $Seg_1 = V_{\tau_s:\tau_e}$ in this video, where τ_s, τ_e represent the start and end time of this segment respectively. And in the temporal context of Seg_1, another video segment $Seg_2 = V_{\tau_s':\tau_e'}$ is obtained by randomly shifting along the temporal axis, where $\tau_s' \in [\tau_s - \varepsilon, \tau_s + \varepsilon]$, $\tau_e' \in [\tau_e - \varepsilon, \tau_e + \varepsilon]$. The hypothesis of temporal context consistency is that the predicted category distributions for these two adjacent segments Seg_1 and Seg_2 are similar, if the spatio-temporal information of the segments have been adequately seen by the network during training. Otherwise, if there are false predictions between these two segments, the divergence between the predicted category distributions of Seg_1 and Seg_2 will be high. When observing the whole time steps, the high divergence between adjacent time steps will lead to temporal context inconsistency. Thus, by computing the divergence of predictions locally, we will be able to approximate the degree to which the prediction of a temporal proposal in the corresponding location is incorrect.

For each video sample V_i in the unlabeled pool χ_U^0, the predicted category distribution of temporal proposal corresponding to the t-th temporal location is $\tilde{y}_{i,l,t} \in \mathbb{R}^C$, the distribution of all proposals is $\tilde{y}_{i,l} \in \mathbb{R}^{T_l \times C}$. Our aim is to find how temporally-inconsistent these adjacent proposals are from each other, based on the divergence computed in a local context centered at the t-th time step. Firstly, action proposal corresponding to the t-th time step is considered as anchor proposal. With the reference to the local context, the expected category distribution temporally is computed as:

$$\tilde{y}_{i,l}^e = \frac{1}{2r+1} \sum\nolimits_{j=t-r}^{t+r} \tilde{y}_{i,l,j}, \tag{5}$$

where r represents the radius for shift on temporal axis, the higher r indicates the wider range of local temporal context. The frame-level score at the t-th time step is then obtained as follows:

$$s_i[t] = \mathbb{H}\left(\tilde{y}_{i,l}^e\right) - \frac{1}{2r+1} \sum\nolimits_{j=t-r}^{t+r} \mathbb{H}\left(\tilde{y}_{i,l,j}\right), \tag{6}$$

where $\mathbb{H}(p) = -\sum p_i \log(p_i)$, we then have:

$$s_i[t] = -\sum_{c=1}^{C} \left(\tilde{y}_{i,l}^e\right) \log\left(\tilde{y}_{i,l}^e\right) + \frac{1}{2r+1} \sum_{j=t-r}^{t+r} \sum_{c=1}^{C} \tilde{y}_{i,l,t}^c \log\left(\tilde{y}_{i,l,t}^c\right). \tag{7}$$

Through the difference between the entropy of average predictions and the mean entropy of predictions, this score mainly indicate the mutual information between the predicted category distribution of anchor proposal and that of its neighbors. It turns out that $s_i[t]$ will be high if temporal inconsistency occurs in the context of t-th time step.

E. Score Aggregation

The obtained frame-level score $s_i[t]$ described above implies the informativeness of action instance proposal in each temporal location. However, $s_i[t]$ is impractical to rank the unlabeled video samples and decide which one would be manually annotated. Given a video sample V_i, it is appropriate to use a scalar s_i to represent the probability to be annotated. To this end, we specifically consider three functions for aggregating the frame-level score $s_i[t]$ to a scalar s_i.

Maximum frame-level scores. These scores are then max-pooled along the temporal axis:

$$s_i = \max_{t \in [1, T_l]} s_i[t]. \tag{8}$$

Sum frame-level scores. These scores are then summarized:

$$s_i = \sum_{t \in [1, T_l]} s_i[t]. \tag{9}$$

Mean-Max frame-level scores. We equally divide the temporal scale T_l into several non-overlapping regions, and adopt mean-max operation on the frame-level scores. Specifically, the number of regions is denoted as N_R, the maximum frame-level score of the k-th region R_k is firstly computed, and then these local maximum scores are averaged.

$$s_i = \frac{1}{N_R} \sum\nolimits_{k=1}^{N_R} \max_{t \in R_k} s_i[t]. \tag{10}$$

Algorithm 1 Active Learning For Semi-supervised Temporal Action Localization

Input: Initial set of labeled videos χ_L^0,
Acquisition pool of unlabeled videos χ_U^0,
Initialized TAL model Φ_0,
Score aggregation function φ,
Number of active learning cycles M,
Number of query samples in each cycle k.
Output: A well-trained TAL model $\boldsymbol{\Phi_M}$.
for current cycle $m \in [0, M]$ **do**
  Use the labeled data χ_L^m to train model $\boldsymbol{\Phi_m}$.
  for each video sample $V_i \in \chi_U^m$ **do**
    Classify and regress action instances $\tilde{y}_i, \tilde{\tau}_i^s, \tilde{\tau}_i^e \leftarrow \boldsymbol{\Phi_m}(V_i)$.
    Compute frame-level score $s_i[t]$ of each temporal location by TPE/TCI, as described on Equation 4/7.
    Aggregate frame-level scores $s_i \leftarrow \varphi(s_i[t])$, $t \in (1, 2, \cdots, T_l)$.
  end for
  Sort the unlabeled videos in χ_U^m according to video-level scores s_i.
  Select top-k score set S_{top-k}.
  for each video sample $V_i \in \chi_U^m$ **do**
    if s_i is in S_{top-k} **then**
      Annotate video sample V_i.
    else
      Skip and evaluate next sample.
    end if
  end for
  Finish sample selection $\chi_A^m \leftarrow \{V_i\}_{i=1}^k$.
  Update Set $\chi_L^{m+1} \leftarrow \chi_L^m \cup \chi_A^m$, $\chi_U^{m+1} \leftarrow \chi_U^m \setminus \chi_A^m$.
end for

IV. Experiments and results

In this section, we firstly describe experimental settings for a fair comparison in Section IV-A. Then, the comparison with the state-of-the-art methods is demonstrated in Section IV-B. Finally, we show the results of ablation study in Section IV-C.

A. Experimental settings

Dataset. To validate the effectiveness of our method, we conduct extensive experiments on commonly-used benchmark datasets, THUMOS'14 [47] and ActivityNet 1.3 [48]. THUMOS'14 originally comes from the challenge on temporally untrimmed videos held in 2014, and consists of 200 validation videos and 212 testing videos from 20 categories labeled for temporal localization. In our experiments, videos are split into 7982 samples for training, each sample consists of an action instance. ActivityNet 1.3 has 19994 videos with 200 action classes. This dataset is divided into training, validation and testing subset by 2:1:1.

Metrics. We utilize two metrics for evaluating our proposed method respectively. Given the *fixed label budget* (*i.e.*, the fixed labeled training samples), we evaluate the localization performance of models trained with different amount of video samples in different cycles. Given the *fixed localization performance*, we evaluate the label cost of models for training.

We take mean Average Precision (mAP) at certain IoU thresholds as the main localization metric. The thresholds are $[0.3, 0.4, 0.5, 0.6, 0.7]$ for THUMOS'14 and $[0.5, 0.75, 0.95]$ for ActivityNet 1.3. Relative Saving (RS) of label cost is used to evaluate the capability of proposed method about reducing label cost. When achieving the same localization performance, N_A labeled samples are needed in method A, while N_B labeled samples are needed in method B, the RS value of N_A compared to N_B can be computed as follows:

$$RS = \frac{N_B - N_A}{N_B} \times 100\%. \tag{11}$$

As shown in Equation 11, $RS > 0$ indicates that fewer label cost is needed in method A than that in method B. The higher RS value is, the larger label cost will be saved.

Competitors. In addition to the self-comparison experiments, our proposed AL-STAL is compared with the state-of-the-art unsupervised method (ACL) and semi-supervised methods (SSTAP, MTMD, KFC, Semi-AFSD) with only 10% or 60% *randomly-selected* labeled data.

- ACL [49] Without any manual annotation, the ACL approach defined a "cluster-localize" pipeline to generate pseudo labels for training a TAL model.
- SSTAP [22]. The SSTAP method designed different temporal sequential perturbations and defined classification, temporal order prediction, consistency constraints for generating action proposals to be localized.
- MTND [23]. The MTND method utilized time warping and time masking as sequential perturbations and developed a novel framework based on Mean Teacher [50].
- KFC [21]. The KFC approach adopted the K-farthest crossover as feature perturbations and used supervised classification loss and consistency loss simultaneously for training.
- Semi-AFSD [7]. The Semi-AFSD method directly trained the original AFSD model with the randomly-selected labeled data.

Also, the adopted base action localizer in AL-STAL (*i.e.*, AFSD) with full annotations are served to provide an upper bound performance for our learning scenario. To be specific, we directly train the action localization network with full training set (equivalent to reproduction of AFSD) and report the performance of AFSD as the **Oracle** for comparison. Even though most competitors conducted experiments on both datasets, there are a few competitors which only reported their results on one dataset. For fair comparison, we follow most of competitors and use I3D feature.

Implementation details. For base TAL model, we follow the former setting of AFSD. On THUMOS'14, both RGB and optical flow frames are sampled at 10 frames per second (fps) and videos are split into clips. The length of each clip T is set as 256 frames. Adjacent clips have a temporal overlap of 30 frames in training and 128 frames in testing. On ActivityNet1.3, we sample frames using different fps and ensure the number of video frames is 768 for each video. We finetune a pre-trained I3D [44] model, and the whole localization model is trained for 25 epoches using Adam [51] with learning rate of 10^{-5}, weight decay of 10^{-3}. Batchsize is set to 1, the weight of loss λ is set to 10 on THUMOS'14 and 1 on ActivityNet 1.3 and γ is set to 1 on both datasets. In the testing phase, the results of RGB and optical flow frames are averaged to obtain final predictions.

For AL-STAL method, we use the same settings on THUMOS'14 and ActivityNet 1.3. k is set to $0.05 \times |\chi|$, where $|\chi|$ is the number of samples in the whole dataset, *i.e.* only 5% unlabeled samples are selected in each cycle. The initial number of samples in χ_L^0 is also $0.05 \times |\chi|$ for both datasets. In the Mean-Max score aggregation, we set N_R as 4, four temporal regions are utilized for aggregation.

B. Comparison with the State-of-the-art

We compare our active learning method with the state-of-the-art methods and report the localization performance and labeled data used by each method.

TABLE I
mAP(%) RESULTS WHEN COMPARING WITH STATE-OF-THE-ART SEMI-SUPERVISED AND UNSUPERVISED METHODS ON THUMOS'14.

Method	Reference	THUMOS'14					
		0.3	0.4	0.5	0.6	0.7	Avg.
0% labeled data							
ACL [49]	CVPR'20	36.9	32.9	25.0	16.7	8.9	24.1
10% labeled data							
MTND [23]	ICCV'19	43.4	34.4	25.8	17.4	9.5	26.1
SSTAP [22]	CVPR'21	45.5	35.9	27.6	18.4	11.1	27.7
AL-STAL@10%	Ours	**66.1**	**60.6**	**51.9**	**39.8**	**25.3**	**48.8**
60% labeled data							
MTND [23]	ICCV'19	53.4	45.2	37.2	29.5	20.5	37.2
SSTAP [22]	CVPR'21	56.5	48.8	39.4	30.5	20.7	39.2
KFC [21]	TIP'21	57.7	51.5	43.3	32.4	22.9	41.6
AL-STAL@60%	Ours	**65.2**	**60.4**	**52.3**	**41.6**	**28.0**	**50.3**
Oracle	-	67.3	62.4	55.5	43.7	31.1	52.0

Comparison on THUMOS'14. Table I compares our active learning method with the state-of-the-art semi-supervised and unsupervised action localization methods on THUMOS'14. We report mAP at different tIoU thresholds, as well as average mAP. When using 10% labeled data, our AL-STAL obtains a score of (66.1%, 60.6%, 51.9%, 39.8%, 25.3%,

48.8%) for the metrics mAP at different IoU thresholds, and reaches an average mAP of 48.8%. When using 60% labeled data, our AL-STAL outperforms the competitors MTND, SSTAP and KFC by a large margin, the improvements of average mAP are 13.1%, 11.1% and 8.7% respectively when comparing with these competitors. The competitors can only achieve limited performance as they overlook the discrepancy in informativeness between video samples. The consistent advantages when using 10% and 60% labeled data indicate that our proposed active learning method is able to improve localization performance through continuously selecting the highly-informative samples for annotation. It is worth noting that when the amount of labeled data decreases from 60% to 10%, there is a significant drop of average mAP in both MTND and SSTAP method (37.2% $\rightarrow$ 26.1%, 39.2% $\rightarrow$ 27.7%). Compared with MTND and SSTAP, there is a minor gap between the mAP performance of our AL-STAL with 60% and 10% labeled data (50.3% $\rightarrow$ 48.8%), indicating that AL-STAL is relatively insensitive to the decrease of labeled data. The reason is that the 10% labeled samples in AL-STAL are more informative, thus most of inherent information for supervision is maintained, even though the amount of labeled sample is sharply decreased from 60% to 10%.

When comparing with unsupervised method ACL, AL-STAL enjoys a significant mAP improvement only using 10% labeled data (48.8% *vs.* 24.1%), while the improvements of MTND and SSTAP are evidently lower (26.1% *vs.* 24.1%, 27.7% *vs.* 24.1%). This significant improvement shows our AL-STAL has favorable potential when label information is extremely insufficient. Due to the different percentages of labeled data for training, there is an inevitable gap between the results of AL-STAL and the Oracle method (1.7% in average mAP). However, it should be noted that the gap is relatively slight when saving 40% label cost in the meantime.

TABLE II
mAP(%) RESULTS WHEN COMPARING WITH STATE-OF-THE-ART SEMI-SUPERVISED AND UNSUPERVISED METHODS ON ACTIVITYNET 1.3.

Method	Reference	ActivityNet			
		0.5	0.75	0.95	Avg.
0% labeled data					
ACL [49]	CVPR'20	35.2	21.4	3.1	21.1
10% labeled data					
MTND [23]	ICCV'19	38.9	28.7	8.4	27.6
SSTAP [22]	CVPR'21	40.7	29.6	9.0	28.2
AL-STAL@10%	Ours	**48.8**	**31.3**	**4.3**	**31.1**
60% labeled data					
Semi-AFSD [7]	CVPR'21	49.0	31.4	5.9	31.2
MTND [23]	ICCV'19	49.8	34.5	7.0	33.5
SSTAP [22]	CVPR'21	50.1	**34.9**	7.4	34.0
KFC [21]	TIP'21	**51.6**	**34.9**	**9.0**	**34.4**
AL-STAL@60%	Ours	**50.3**	**34.7**	**7.8**	**34.1**
Oracle	-	52.4	35.3	6.5	34.4

Comparison on ActivityNet 1.3. Like Table I, we conduct experiments on ActivityNet 1.3 for comparison using 10% and 60% labeled data and report the results in Table II. As shown in Table II, our AL-STAL outperforms all the competitors significantly when using 10% data, and achieves better or comparable performance when expanding the labeled set into 60%. Specifically, AL-STAL achieves (48.8%, 31.3%, 4.3%) for the metrics mAP at different IoU thresholds, and reaches an average mAP of 31.1% when using 10% labeled data. Compared with MTND and SSTAP, the average mAP results are improved by 3.5% and 2.9% respectively. When using 60% labeled data, AL-STAL surpasses the Semi-AFSD that directly trains the AFSD model with randomly-selected samples. Also, better performance is obtained when compared with MTND and SSTAP. When comparing with KFC, our proposed AL-STAL performs slightly inferior to this method (34.1% *vs.* 34.4%). The reason is that KFC constructs complicated perturbations for model input, and employs additional consistency regularization. The heavy pre-process on features turns KFC into complicated for practical applications. Given its simplicity, AL-STAL achieves comparable performance on ActivityNet 1.3, making AL-STAL worthy of further analysis. Notably when decreasing the amount of labeled data from 60% to 10%, only 3% average mAP is dropped in AL-STAL, while the performance drops of MTND and SSTAP are 5.9% and 5.8% respectively. This result again demonstrates that our proposed AL-STAL is able to harness the informative samples and reserve the supervision information mostly.

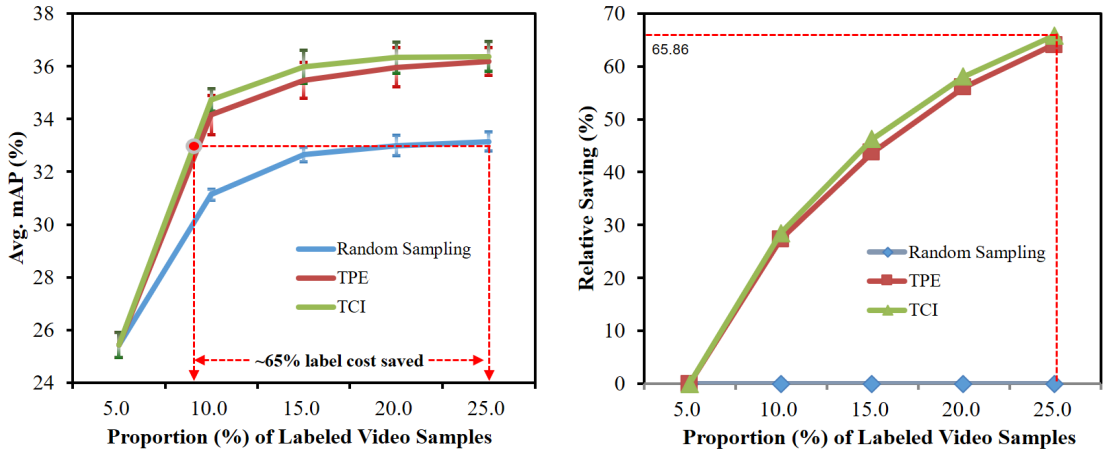
Besides, a superior performance can be found when comparing AL-STAL with the unsupervised advanced method ACL. Only using 10% labeled data, the mAP results are increased from (35.2, 21.4, 3.1, avg. 21.1) to (48.8, 31.3, 4.3, avg. 31.1). While MTND and SSTAP achieve limited success in boosting performance using the same amount of labeled data. At last, it can be found that there is a quite slight gap between AL-STAL and the Oracle method (only 0.3% average mAP). The comparable performance with only 60% labeled data proves that our method is able to maintain localization performance and meanwhile reduce large amounts of label cost.

C. Ablation Studies

To verify the effectiveness of the proposed method, we conduct ablation studies with the RGB model and provide the experiment results as follows, including the choice of scoring function and unlabeled pool for sample selection. Also, optical flow model is involved for discussion.

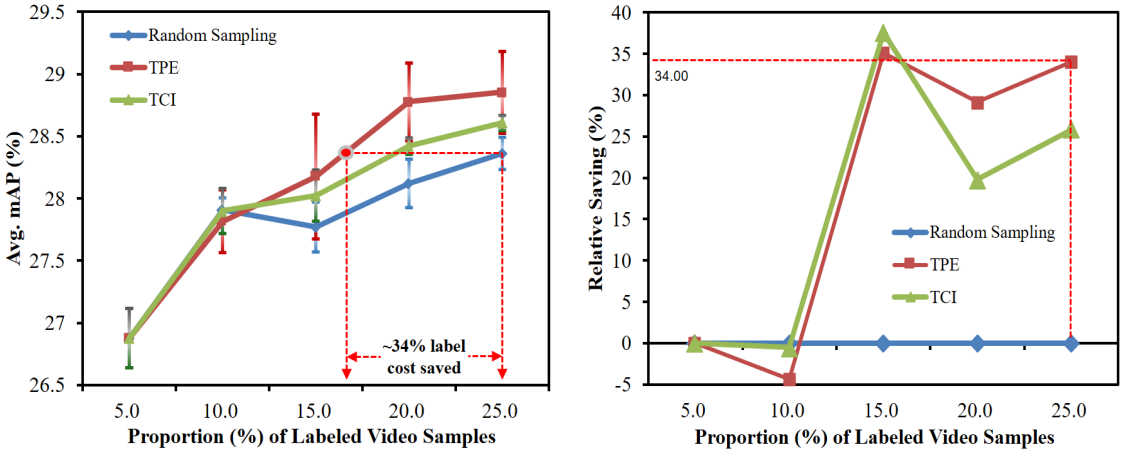
Our aim is to select informative samples for annotating, rather than randomly selecting labeled data. Thus, the baseline method and proposed methods for ablation are as follows:

- **Rand.** We choose Random Sampling strategy (Rand.) as our baseline for validation. According to Random Sampling, video samples in the unlabeled pool χ_U are randomly selected and annotated in each cycle. Nevertheless, the significant differences between the informativeness of samples are neglected, and the selection scores corresponding to these samples are randomly initialized.
- **TPE.** To evaluate the effectiveness of TPE, we replace the Random Sampling strategy with TPE-based sample selection, which is defined in Section III-C.
- **TCI.** In the process of selecting video sample for annotation, we replace the Random Sampling strategy with TCI-based sample selection defined in Section III-D.



(a) Average mAP results (b) Relative label cost saving

Fig. 6. **(a)** Mean average precision curve of scoring functions for sample selection on THUMOS'14 dataset. Each point in the plot is an average of 5 trials. **(b)** Relative label cost saving of different sample selection methods on THUMOS'14 dataset.



(a) Average mAP results (b) Relative label cost saving

Fig. 7. **(a)** Mean average precision curve of scoring functions for sample selection on ActivityNet 1.3 dataset. Each point in the plot is an average of 3 trials. **(b)** Relative label cost saving of different sample selection methods on ActivityNet 1.3 dataset.

Effectiveness of TPE and TCI. For fair comparison, we adopt the same initialized model and labeled data at the initial cycle, and the only variable in ablation experiments is the scoring function in sample selection.

Fixed label budget. Fig. 6(a) and Fig. 7(a) show the mAP performance with different scoring functions in each cycle on both datasets respectively, the proportion of labeled data for training is from 5% to 25%, and each mAP result is obtained by average values in multiple trials. We start with analysis of experiment results on THUMOS'14, as shown in Fig. 6(a). When comparing with baseline method of random sampling, our proposed methods achieve significant improvements. We observe that starting from the first two cycles, TPE and TCI outperform the baseline by 3.03% and 3.59% respectively, and the performance improvement maintains in the other active learning cycles. In the last one, where we actively annotate 25% of training samples, TPE and TCI can have 3.04% and 3.22% average mAP improvement respectively, which supports our claim that the proposed scoring functions is verified to be effective in labeled dataset update. Given fixed label budget, our method is able to boost the localization performance by prioritizing labeling the informative video samples. On ActivityNet 1.3, although fluctuation occurs in the second cycle, we are able to get the same observation of proposed method from Fig. 7(a). TPE and TCI perform better for sample selection in most of the cases when comparing with baseline method, which again demonstrates the effectiveness of our proposed method. Finally, specific results on both datasets when using 25% labeled training in the last cycle are shown in Table III, and our proposed method outperforms the baseline at all the used tIoU thresholds.

Fixed mAP performance. On THUMOS'14, we report the RS value with different scoring functions in each cycle, the proportion of labeled data for training is from 5% to 25%. Given the mAP performance of baseline method in each cycle, the corresponding label cost and RS value of proposed method

TABLE III
mAP(%) RESULTS WHEN USING 25% LABELED TRAINING DATA ON BOTH THUMOS'14 AND ACTIVITYNET 1.3 DATASETS AT ALL THE USED TIOU THRESHOLDS.

Selection Methods	THUMOS'14					ActivityNet		
	0.3	0.4	0.5	0.6	0.7	0.5	0.75	0.95
Rand.	48.63	43.10	35.09	24.92	14.05	46.32	28.82	3.35
TPE	**53.06**	**47.72**	39.11	28.06	15.85	**46.63**	**29.34**	**4.54**
TCI	52.67	47.33	**39.75**	**29.15**	**16.55**	46.22	28.78	3.97

could be inferred with the mAP curve. Likewise, the RS results are obtained by average values in 5 trials. As shown in Fig. 6(b), TPE and TCI methods have large positive relative saving of label cost when comparing with baseline, indicating AL-STAL can achieve comparable localization performance with fewer labeled data. Besides, the RS value increases as the labeled data enriches from 5% to 25%, showing that our AL-STAL method is able to save label cost continuously as training set updates. On ActivityNet 1.3, Fig. 7(b) shows that the RS value of proposed methods are positive in most of cases, which confirms the efficacy of our proposed method. Specifically, Table IV demonstrates the label cost relative saving when using 25% labeled training data. It is noteworthy that when equipped with TPE and TCI methods, over 60% and 30% annotation works are saved on the two datasets respectively, which fulfills our expectation of reducing label cost for temporal action localization.

TABLE IV
RELATIVE SAVING (RS) OF LABEL COST (%) WHEN USING 25% LABELED TRAINING DATA ON BOTH THUMOS'14 AND ACTIVITYNET 1.3 DATASETS.

Selection Methods	RS Value	
	THUMOS'14	ActivityNet
Rand.	0.0	0.0
TPE	64.16	34.00
TCI	65.86	25.83

Instantiation of score aggregation. Frame-level scores $s_i[t], t \in (1, 2, \cdots, T_l)$ need to be aggregated into a scalar s_i to represent the probability of samples to be manually annotated. Here, we compare our proposed score aggregation function with the following three instantiations: *Max*, *Sum* and *Mean-max* on THUMOS'14. Likewise, we increase the proportion of labeled video samples from 5% to 20%, and present the average mAP results in Table V. Among all instantiations, TPE with *Sum* aggregation receives the best performance, achieving 26.21%, 34.81%, 36.41%, 36.75% respectively. TCI is in good coordination with the *Max* aggregation, reaching 26.21%, 34.77%, 36.19% and 36.43%.

RGB and Flow stream. In addition to the experiments with RGB data stream, optical flow stream also serves as the input of our model, and then both results of these two inputs are fused for final prediction in this experiment. The results on both datasets with different modalities are shown in Table VI. Similar to RGB model, evident improvements could be obtained when replacing random sampling with our proposed TPE and TCI. Specifically when given 25% labeled data, TPE offers 3.22% gains in average mAP as compared to using random sampling, and TCI attains 2.94% improvement. Besides, the localization performance could be further boosted with multi-modal fusion. Using 25% labeled data, the average mAP of TPE and TCI method are reaching 44.39% and 44.74% respectively. The significant improvements in RGB, optical flow and multi-modal fusion results again demonstrate the effectiveness of our proposed AL-STAL method.

TABLE V
AVG. mAP(%) RESULTS WITH DIFFERENT SCORE AGGREGATIONS ON THUMOS'14 DATASET.

Selection		Score Aggregation			Proportion of labeled samples			
TPE	TCI	Max	Sum	M-max	5.0%	10.0%	15.0%	20.0%
✓		✓			26.21	34.75	35.67	36.18
✓			✓		26.21	**34.81**	**36.41**	**36.75**
✓				✓	26.21	33.42	35.39	36.45
	✓	✓			26.21	**34.77**	**36.19**	**36.43**
	✓		✓		26.21	33.96	35.37	36.02
	✓			✓	26.21	34.56	36.01	36.42

TABLE VI
AVG. mAP(%) RESULTS WITH DIFFERENT MODALITIES ON THUMOS'14 DATASET.

	Modality		Proportion of labeled samples				
	RGB	Flow	5.0%	10.0%	15.0%	20.0%	25.0%
Rand.	✓		25.44	31.13	32.65	32.99	33.15
		✓	26.57	31.08	32.49	33.10	33.32
	✓	✓	34.03	39.25	40.83	41.71	42.40
TPE	✓		25.44	34.16	35.46	35.97	36.19
		✓	26.57	33.91	35.22	36.33	36.54
	✓	✓	**34.03**	**42.33**	**43.62**	**43.90**	**44.39**
TCI	✓		25.44	34.72	35.99	36.33	36.37
		✓	26.57	34.46	35.78	36.23	36.26
	✓	✓	**34.03**	**43.12**	**44.21**	**44.59**	**44.74**

V. CONCLUSION AND FUTURE WORK

In this paper, we focus on a rarely investigated yet practical problem of semi-supervised temporal action localization and proposes an effective active learning method for this task, named AL-STAL. We address the motivation of only selecting highly-informative samples for annotation, and present two scoring functions which facilitate the video sample rank and selection based on the uncertainty of localization model. TPE takes entropy of predicted label distribution as measure of uncertainty, and TCI introduces a new metric based on mutual information between adjacent action proposals along the temporal axis. Experiment results on THUMOS'14 and ActivityNet 1.3 demonstrate the effectiveness and superiority of our proposed method. Furthermore, we consider that a comprehensive system including active learning, weakly-supervised learning and semi-supervised learning would benefit the larger reduction of label cost in practice. In our

future plan, active learning would incorporate multi-level supervisions ($e.g.$ video-level and instance-level) derived from annotators, and the labor efforts would be continually saved.

VI. Acknowledgments

The authors would like to thank editor and anonymous reviewers.

References

[1] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, June. 14, 2022. doi: 10.1109/TPAMI.2022.3183112

[2] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li, "A comprehensive study of deep video action recognition," 2020, *arXiv:2012.06567*. [Online]. Available: https://arxiv.org/abs/2012.06567

[3] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[4] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 3889–3898.

[5] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7094–7103.

[6] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, and R. Ji, "Fast learning of temporal action proposal via dense boundary generator," in *Proc. 34th Conf. Artif. Intell. (AAAI)*, 2020, pp. 11 499–11 506.

[7] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Learning salient boundary feature for anchor-free temporal action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 3320–3329.

[8] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 5783–5792.

[9] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 344–353.

[10] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM International Conference on Multimedia (MM)*, 2017, pp. 988–996.

[11] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1130–1139.

[12] H. Alwassel, F. C. Heilbron, and B. Ghanem, "Action search: Spotting actions in videos and its application to temporal action localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 251–266.

[13] H. Song, X. Wu, B. Zhu, Y. Wu, M. Chen, and Y. Jia, "Temporal action localization in untrimmed videos using action pattern trees," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 717–730, 2018.

[14] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 5734–5743.

[15] J. Gao and C. Xu, "Learning video moment retrieval without a single annotated video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1646–1657, 2021.

[16] B. Wang, X. Zhang, and Y. Zhao, "Exploring sub-action granularity for weakly supervised temporal action localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2186–2198, 2022.

[17] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 154–171.

[18] X. Ding, N. Wang, J. Li, and X. Gao, "Weakly supervised temporal action localization with segment-level labels," in *Proc. Springer Conf. Pattern Recognit. Comput. Vis. (PRCV)*, 2021, pp. 42–54.

[19] G. Li, J. Li, N. Wang, X. Ding, Z. Li, and X. Gao, "Multi-hierarchical category supervision for weakly-supervised temporal action localization," *IEEE Trans. Image Process.*, vol. 30, pp. 9332–9344, 2021.

[20] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization." 2019, *arXiv:1911.09963*. [Online]. Available: https://arxiv.org/abs/1911.09963

[21] X. Ding, N. Wang, X. Gao, J. Li, X. Wang, and T. Liu, "Kfc: An efficient framework for semi-supervised temporal action localization," *IEEE Trans. Image Process.*, vol. 30, pp. 6869–6878, 2021.

[22] X. Wang, S. Zhang, Z. Qing, Y. Shao, C. Gao, and N. Sang, "Self-supervised learning for semi-supervised temporal action proposal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 1905–1914.

[23] J. Ji, K. Cao, and J. C. Niebles, "Learning temporal action proposals with fewer labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 7073–7082.

[24] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2591–2600, 2016.

[25] M. Rong, H. Cui, Z. Hu, H. Jiang, H. Liu, and S. Shen, "Active learning based 3d semantic labeling from images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, early access, May. 13, 2022. doi: 10.1109/TCSVT.2021.3079991

[26] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 2372–2379.

[27] D. Roth and K. Small, "Margin-based active learning for structured output spaces," in *Proc. Eur. Conf. Machine Learning (ECML)*, 2006, pp. 413–424.

[28] W. Luo, A. Schwing, and R. Urtasun, "Latent structured active learning," in *Proc. Adv. Neural Inf. Process. Syst.(NeuIPS)*, 2013, pp. 728–736.

[29] S. Roy, A. Unmesh, and V. P. Namboodiri, "Deep active learning for object detection," in *Proc. Bri. Mach. Vis. Conf. (BMVC)*, 2018, pp. 91–103.

[30] Y. Huang, Q. Dai, and Y. Lu, "Decoupling localization and classification in single shot temporal action detection," in *Proc. IEEE Inter. Conf. Multimedia Expo (ICME)*, 2019, pp. 1288–1293.

[31] C. Jin, T. Zhang, W. Kong, T. Li, and G. Li, "Regression before classification for temporal action detection," in *Proc. IEEE Int. Conf. Acous. Speech Signal Proce. (ICASSP)*, 2020, pp. 1–5.

[32] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3628–3636.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.(NeuIPS)*, 2015, pp. 91–99.

[34] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–40, 2021.

[35] B. Settles, "Active learning literature survey," Tech. Rep., 2009. [Online]. Available: https://minds.wisconsin.edu/handle/1793/60660

[36] M. Prince, "Does active learning work? a review of the research," *Journal of engineering education*, vol. 93, no. 3, pp. 223–231, 2004.

[37] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process (EMNLP)*, 2008, pp. 1070–1079.

[38] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2016, pp. 1050–1059.

[39] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 93–102.

[40] H. H. Aghdam, A. Gonzalez-Garcia, J. v. d. Weijer, and A. M. López, "Active learning for deep detection neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 3672–3680.

[41] I. Elezi, Z. Yu, A. Anandkumar, L. Leal-Taixe, and J. M. Alvarez, "Towards reducing labeling cost in deep object detection," 2021, *arXiv:2106.11921*. [Online]. Available: https://arxiv.org/abs/2106.11921

[42] H. S. Hossain, M. A. Al Haiz Khan, and N. Roy, "Deactive: scaling activity recognition with active deep learning," in *Proc. ACM Inter., Mobile, Wearable Ubiq. Tech.*, vol. 2, no. 2, 2018, pp. 1–23.

[43] H. Wang, X. Chang, L. Shi, Y. Yang, and Y.-D. Shen, "Uncertainty sampling for action recognition via maximizing expected average precision," in *Proc. 27th Int. Joint Conf. Artif. Intell (IJCAI)*, 2018, pp. 964–970.

[44] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 6299–6308.

[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2980–2988.

[46] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 5561–5569.

[47] Y. G. Jiang, J. Liu, R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes." 2014. [Online]. Available: https://www.crcv.ucf.edu/THUMOS14/

[48] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 961–970.

[49] G. Gong, X. Wang, Y. Mu, and Q. Tian, "Learning temporal co-attention models for unsupervised video action localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 9819–9828.

[50] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst. (NeuIPS)*, 2017, pp. 1195–1204.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, arXiv:1412.6980. [Online]. Available: https://arxiv.org/abs/1412.6980