Learning a Robust Model with Pseudo Boundaries for Noisy Temporal Action Localization

Xinyi Yuan
University of Science and Technology of China
Hefei, Anhui, China
xinyiyuan@mail.ustc.edu.cn

Liansheng Zhuang*
University of Science and Technology of CHina
Hefei, Anhui, China
lszhuang@ustc.edu.cn

ABSTRACT

Temporal Action Localization (TAL) aims to locate starting and ending times of actions and recognize categories in untrimmed videos. Significant progress has been made in developing deep models for TAL. The success of previous methods relies on large-scale training data with precise boundary annotations. However, fully accurate annotations are unpractical to be obtained due to the ambiguities of the action boundaries and the crowd-sourcing labeling process, leading to a degradation in performance. In this work, we take the first step into learning with inaccurate boundaries in TAL tasks. Motivated by the fact that inaccurate boundary annotations harm localization precision more than classification accuracy, we propose to use classification as a guidance signal to improve localization precision. Specifically, we introduce a pseudo-boundary generation and refinement method (PbGaR). PbGaR first treats each action segment as a bag of instances to select the instances with more accurate boundaries for training. Then these boundaries are refined via two strategies for higher quality. The proposed method significantly alleviates the degraded performance of TAL models under inaccurate boundaries. Extensive experiments on two popular datasets demonstrate the effectiveness of our method.

CCS CONCEPTS

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

KEYWORDS

Video understand, temporal action localization, inaccurate boundaries

ACM Reference Format:
Xinyi Yuan and Liansheng Zhuang. 2023. Learning a Robust Model with Pseudo Boundaries for Noisy Temporal Action Localization. In *ACM Multimedia Asia 2023 (MMAsia '23), December 06–08, 2023, Tainan, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3595916.3626455

1 INTRODUCTION

Due to its wide applications in surveillance, video retrieval [5] and video anomaly detection [23], the task of Temporal Action Localization (TAL) has drawn much attention in the computer vision communities. Remarkable progress has been made under the fully-supervised setting in recent years. Under this setting, the success of previous methods rely on large-scale video datasets like ActivityNet-1.3 [1] and THUMOS14 [9] with precise boundary annotations. However, fully accurate annotations are unpractical to be obtained in professional fields, thus limiting scalability and practicability in real-world scenarios. In TAL, noisy annotations refer to inaccurate categories and boundaries. However, inaccurate boundaries is more common than that of categories'. As in some domains such as sports competitions, the labeling of the start and the end of an action is strict and difficult. One can recognize most action categories by key frames alone, but needs to browse through all frames to get the accurate boundaries of an action. However, many annotators lack of expertise, leading to inaccurate action boundary annotations. Besides, with the increasing video data, many datasets are annotated by crowd-sourcing or volunteers within limited budgets and resources [30]. This undoubtedly results in low-quality annotations and further affects the training process of existing models. Ultimately, this cause performance degradation of models. In view of these phenomena, handling inaccurate annotations especially inaccurate boundaries is a critical and pressing task.

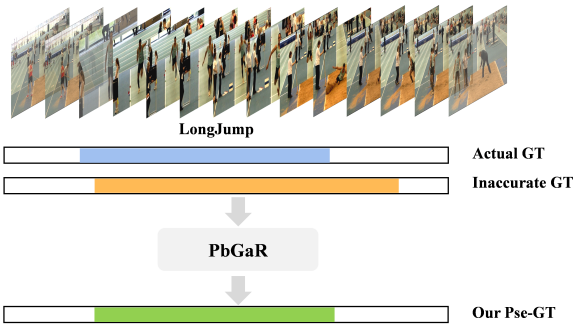

Figure 1: Inaccurate boundary annotation illustration. TAL methods suffer from inaccurate annotations. Our method PbGaR generates a more precise boundary for training.

There are two main types of frameworks in temporal action localization, i.e. two-stage method and one-stage method. Two-stage method generate candidate proposals at first, then take strategies to recognize categories among proposals and further refine the predictions. Recently, one-stage method has become the mainstream due to its simplicity and efficiency. Such method classifies and localizes actions simultaneously. Despite these facilitative work in TAL, the quality of supervised learning models depends on the quality of training datasets [15]. Inaccurate annotations can directly mislead

*Corresponding author.

MMAsia '23, December 06–08, 2023, Tainan, Taiwan

ACM ISBN 979-8-4007-0205-1/23/12.
https://doi.org/10.1145/3595916.3626455

models to learn or memorize wrong relations and thus limit the abilities of these models and deteriorate the performance. No work before has considered the impact of noisy annotations on TAL model performance. In this work, for the first time, we step forward the temporal action localization with noisy annotations. Considering the delicate requirements for boundary detection on TAL tasks and the reality that boundary labeling is more error-prone, we focus on tackling inaccurate boundary annotations.

Motivated by weakly-supervised TAL [7, 27] and object detection [14], we propose a method to improve the degraded performance of TAL models under low-quality boundary annotations. Inspired by the fact that compared with localization, classification precision suffers slightly from inaccurate annotated boundaries, we propose leveraging classification as a guidance signal for localization based on multiple instance learning (MIL) [3]. Specifically, every labeled action is treated as a bag of intances (i.e. a bag of action proposals with same action segment). Our target is to select the most accurate instance from each bag to generate pseudo boundaries and then replace the original inaccurate boundary annotations for training. Our method called PbGaR basically consists of two parts, i.e. pseudo-boundary generation module and pseudo-boundary refinement module. The former is to generate instances with more accurate pseudo-boundaries based on MIL and the latter aims to further enhance the quality of pseudo-boundaries. The proposed mothod can improve the robustness of existing TAL models when dealing with inaccurate noisy boundary annotations. We construct noisy datasets based on two benchmark THUMOS14 and ActivityNet-1.3 and conduct experiments. Extensive experimental results prove the effectiveness of our proposed method.

Our main contributions can be summarized as follow:

- This paper proposes a novel framework for Temporal Action Localization with inaccurate boundary annotations. To our best knowledge, this paper is the first attempt to deal with this setting.
- By carefully generating and refining more accurate pseudo boundaries for training, our proposed method can considerably improve the performance in different degrees of noisy data, thus boost the robustness of existing TAL models.
- Extensive experiments on public benchmarks ActivityNet-1.3 and THUMOS14 show that our proposed method achieves remarkable improvement.

2 RELATED WORK

2.1 Fully-Supervised TAL

Fully-Supervised Temporal Action Localization is a process where temporal boundaries and categories of action instances are available for training. There are mainly two kinds of frameworks in fully-supervised TAL, i.e. one-stage method and two-stage method. Two-stage method generates candidate proposals at first, then take strategies to recognize categories among proposals and further refine the predictions. One-stage method has become the mainstream recently for its simplicity and efficiency. ActionFormer [31], as one of the representative methods , without using proposals, classified every moment into action categories and simultaneously regressing their corresponding boundaries. Additionally, it introduced a Transformer-based [26] network to extract multiscale features, which significantly boosted its performance. TriDet [19] improved on the structure of Transformer and proposed to model relative probability distribution of boundaries, thus going a step further in localization accuracy. The above approaches assume that all of the training data in untrimmed videos are accurate and clean, which impedes their application to real scenario. Our paper takes a rigorous perspective, focuses on how TAL models do their best when facing with inaccurate annotations, especially inaccurate boundaries.

2.2 Weakly-Supervised TAL

Weakly-Supervised Temporal Action Localization is a more resource-efficient setting that has become popular recently. In training process, only video-level classification labels are available. UntrimmedNet [27] firstly introduced Multiple Instance Learning (MIL) [3] to this task. MIL assumes that all instances (i.e. frames in an untrimmed video) belong to a bag that is either positive or negative. In other words, considering a video as a bag consists of frames, MIL-based method would assign the video-level labels on a set of instances (frames). Subsequently, many derivative work [7, 10, 11, 17] followed the MIL-based framework and advanced the development of weakly-supervised TAL. The newest method [17] replaced segment-based MIL framework with proposal-based one to tackle the inconsistent objectives between training and testing stages. Notably, our work differs from weakly-supervised TAL in that we focus on settling TAL models with frame-level annotations rather than being provided with only video-level classification annotations. Although we also formulate TAL as a MIL problem, we regard each action in the video as the concept of bag instead. Besides, our bag can be constructed in a dynamic way to better correct noisy boundaries.

2.3 Learning with noisy data

Work in image domain, especially image classification and object detection, is closely related to video-understanding tasks. There has been a series of studies [8, 12, 15, 28, 30, 33] on noisy data in image tasks. Some methods [18, 21] designed re-weighting strategies to adaptively assign different weights to noisy samples and clean samples. Another major line to minimize the impact of corrupted labels is loss correction. Common methods along this line use a confusion matrix [25], design extra inference steps to correct corrupted labels [6, 16, 22] or replace hard labels with soft labels for unclear boundaries [4, 8, 29]. In addition to the two main directions of resolution mentioned above, Liu *et al.* [14] proposed to correct the inaccurate annotations to facilitate the object detectors in a MIL-based framework. Most existing image tasks focus on noisy classification label, but for video domain, due to the complexity and diversity of action instances, misclassification is considerably less common than localization [15], thus making annotators prone to inaccurate boundaries. In this paper, we are the first to step further towards TAL with inaccurate boundary problem. The uniqueness of localization in untrimmed videos makes this task more challenging and valuable.

3 METHOD

An untrimmed video V can be represented by a set of features $\mathbf{X} = \{x_1, x_2, \ldots, x_T\}$, where T denotes the number of instances. Fully-Supervised TAL consists of two sub-tasks, i.e. classification and

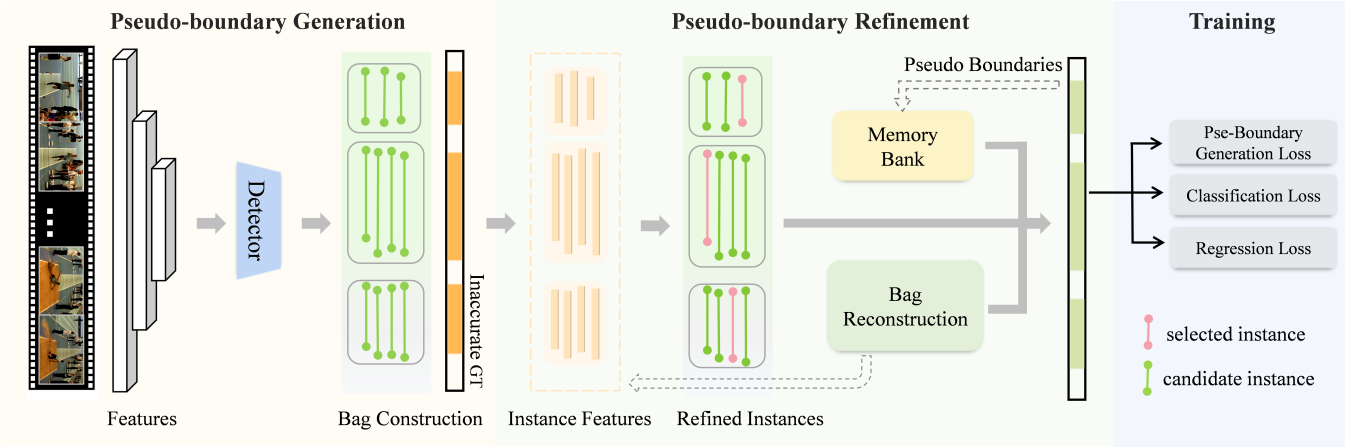

Figure 2: The overall framework of the proposed PbGaR. Features of the untrimmed video are fed into detectors. Based on the output by detector, our pseudo-boundary generation module treats every inaccurate GT (orange blocks) as a bag of instances (green lines). We select the most positive instance (pink lines) to generate the pseudo boundary. A refinement module consisted of two strategies (i.e. bag reconstruction and memory bank) is applied to further enhance the quality of pseudo boundaries (green blocks). Pseudo boundaries will be used for training detectors.

localization. It aims to detect all action labels $Y = \{y_1, y_2, \ldots, y_N\}$ in the video based on the input video features $\mathbf{X}$. Each label $y_i = (s_i, e_i, c_i)$ represents start, end and its action category. During training, every video $\mathbf{V}$ has its segment-level annotations $y = (s, e, c) \in \mathbb{R}^C$, where $\mathbf{C}$ represents the number of ground-truth. Fully-supervised TAL undoubtedly suffers from inaccurate boundary annotations. Inspired by the fact that compared with localization, classification precision suffers slightly from inaccurate annotated boundaries, we propose to use classification branch as a guide to correct mislabeled boundaries. In the following, we elaborate on our method PbGaR for TAL with inaccurate boundary annotations. As illustrated in Figure 2, PbGaR consists of two modules, including pseudo-boundary generation module and multi-step pseudo-boundary refine module. The former utilizes classification as a guide to generate instances with pseudo-boundary that are more accurate than noisy ground-truth ones based on Multiple Instance Learning [3]. The latter is to further improve the quality of pseudo-boundaries by refining and extending action instances.

3.1 MIL-based Pseudo-boundary Generation

Preliminaries. A typical MIL-based method [17, 27] in weakly-supervised TAL treats each video as a bag of instances (frames) and performs feature extraction on it. Then extracted features are used to calculate confidence score for determining whether frames belong to action or background. Formally, for an untrimmed video containing multiple action categories, video-level action labels denoted as $y \in \{0, 1\}^C$ are given. In order to correspond action categories to specific moments, each video is represented as a bag of instances. MIL use classification loss as the signal to choose suitable instances.

Problem Formulation. In our method, unlike the MIL in weakly-supervised TAL, we treat each segment (action or background) in the video as a bag rather than the entire video. A bag is labeled negative only if all instances in it are negative. Put differently, once

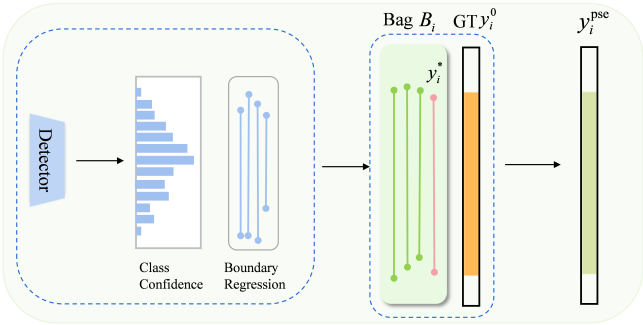

Figure 3: Pseudo-boundary generation module. As the action detectors output class confidence scores and boundary proposals, our first module utilize the output to generate a more accurate pseudo boundary based on MIL.

a instance is positive, the bag would be labeled positive. Formally, let B_i denotes the i^{th} bag in the video V, the j^{th} instance in bag B_i is denoted as y_{ij}. We treat annotated action segments as positive bags B_i^+, instances in it refer to action proposals. Background segments are treated as negative bags B_i^-. The video is formulated as $V = \{B_0^+, B_1^+, \ldots, B_m^+, B_0^-, B_1^-, \ldots, B_k^-\}$. Our goal is to select the most positive instance $y_i^* = \{(s_i^*, e_i^*, c_i)\}$ in the positive bag B_i with unchanged classification label c_i but more precise boundaries $\{s_i^*, e_i^*\}$, i.e. generating a pseudo boundary. We then tab it as a new action annotation for model training. The process of generating y_i^* in bag B_i is denoted as $g(B_i, \theta)$ with parameter θ.

Pseudo-boundary Generation. As in Figure 3, detectors output classification scores and boundary proposals. Based on MIL, we use classification labels predicted by detectors as a guide to build different bags, and instances are action proposals related to the same gt segments. Then we learn to select the most positive instances in each bag. As the initial inaccurate ground-truth boundaries provide

a prior of ambiguous action localization, intuitively, we jointly consider it and the selected action instance from the bag to generate pseudo boundaries for training.

Generally, this module contains two steps. In the first step we obtain the most positive instance y_i^* through $g(B_i, \theta)$. $g(B_i, \theta)$ takes instances in the bag B_i as input and outputs a confidence score in the range of $[-1, +1]$. Index k of y_i^* is obtained from the formula:

$$k = \arg\max g(B_i, \theta). \tag{1}$$

The second step is to generate the final y_i^{pse} by considering the initial action annotation y_i^0 as a complementary. The final new action annotation y_i^{pse} is generated as follows:

$$y_i^{pse} = \omega(g(B_i, \theta)) \cdot y_i^* + (1 - \omega(g(B_i, \theta))) \cdot y_i^0. \tag{2}$$

Here ω is a mapping function that adaptively assigns weights to y_i^* and y_i^0. Considering our goal is to generate as high quality pseudo boundaries as possible, $\omega(\cdot)$ needs to satisfy two conditions. Firstly, as it indicates the confidence of instances, when $g(B_i, \theta)$ outputs large value, higher weight should correspondingly be assigned to y_i^*. Secondly, when $g(B_i, \theta)$ is very close to 1, $\omega(\cdot)$ needs to balance the weight between y_i^* and y_i^0 rather than sharply favoring y_i^*. Thus, a bounded exponential function is adopted to fulfill the two conditions above:

$$\omega(p) = \min(p^\alpha, \beta), \tag{3}$$

where α and β are hyper-parameters and $p \in [0, 1]$.

We adopt a standard hinge loss to train the MIL-based pseudo-boundary generation module. The loss function is defined as:

$$L_{\text{g}}(B_i, \theta) = \max(0, 1 - b_i \max g(B_i, \theta)) \tag{4}$$

$b_i \in 1, -1$ is a label attached to each bag B_i to indicate whether this bag has any positive instance or not.

3.2 Multi-step Pseudo-boundary Refinement

The action instances generated by pseudo-boundary generation module play an important role of new annotations for training detectors. However, they are roughly generated based on the original inaccurate ground-truth annotations and the predictions from the detector. Thus, the quality cannot be guaranteed. Objectively, instances in the same bag have similar properties, i.e. their classification feature and temporal localization are closely related to each other. Besides, our new pseudo boundary is a tradeoff between the initial annotation and the instance selected in the bag by Eq.2. Accordingly, in this section we propose a multi-step refinement module that progressively enhance the quality of the pseudo-boundaries via two strategies.

As we construct the initial bag based on the original ground truth and generate pseudo boundaries, a natural idea arises is that continuing construct new positive bags with these generated pseudo boundaries and repeating the construction until reaching termination condition. This inspires our first strategy, i.e. bag reconstruction strategy. To achieve more efficient reconstruction, we improve the quality of the candidate instances in bags. As illustrated in Figure2, for the j^{th} instance $y_{ij} = (s_j, e_j, c_i)$ in bag B_i, features are sampled at the interval $\{s_j, e_j\}$ via interpolation and aggregated by a fully-connected layer. Boundaries of these instances are then refined and calibrated based on the features. Then we perform bag reconstruction. As in Figure4, new bags are iteratively constructed.

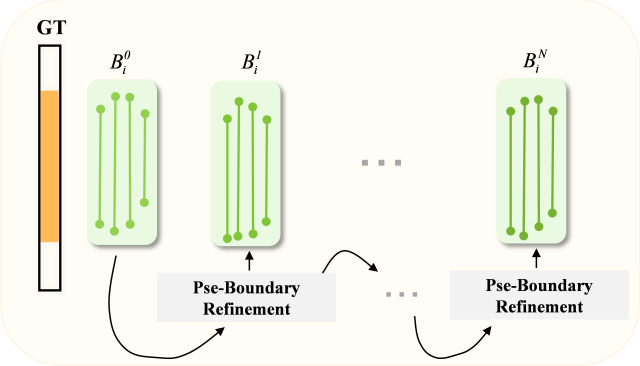

Figure 4: Bag Reconstruction. New positive bags with generated pseudo ground truth are constructed until reaching the termination condition.

After N times iterations of construction, for bag B_i there will be a construction sequence $\{B_i^0, B_i^1, \ldots, B_i^N\}$. Note that negative bags are not involved in this strategy. Consequently, B_i^N is used to optimize the generation module and the loss in Eq.4 is further expressed as:

$$L_{\text{g}}(\{B_i^n, \theta\}) = \sum_n L_{\text{g}}(B_i^n, \theta), \tag{5}$$

where $n \in \{0, 1, \ldots, N\}$ only if bag B_i is positive.

The second strategy called memory bank is to improve the pseudo boundary quality in the generation process described in Eq.(2). After bag reconstruction, pseudo-boundary generation module generates annotations $\{y_0^{pse}, y_1^{pse}, \ldots, y_i^{pse}\}$ with more accurate boundaries for each bag. These annotations are stored in the memory bank and further used with y_i^0 in Eq.2 to provide better localization prior in the next training epoch. Let B_i denote the i^{th} bag, $y_i^{pse(k-1)}$ represents annotations generated in the $k-1^{th}$ epoch. In the k^{th} epoch, we perform a weighted average of $y_i^{pse(k-1)}$ and y_i^0. Therefore, Eq.2 evolves into:

$$\begin{aligned} y_i^{0'} &= \gamma \cdot y_i^0 + (1 - \gamma) \cdot y_i^{pse(k-1)}, \\ y_i^{pse(k)} &= \omega(g(B_i, \theta)) \cdot y_i^* + (1 - \omega(g(B_i, \theta))) \cdot y_i^{0'}. \end{aligned} \tag{6}$$

3.3 PbGaR Training

Training. Our method focuses on providing better performance for TAL detectors in inaccurate training data. It is not limited to specific TAL detectors. In the training stage, as a bootstrap of bag construction, we first train the base detectors (e.g. ActionFormer[31]) for E epochs. Detectors output the probability of action categories and boundary proposals. Based on the output, instances are obtained to construct our initial bags. We adopt an IoU threshold to distinguish which instances are positive. After that, most positive instances are selected to generate pseudo boundaries via Eq.6. Then we apply the pseudo-boundary refinement module to obtain the refined candidate instances with a more accurate estimation of the action location. The same process can be performed with memory bank for multiple steps until the quality of instances is converged, i.e. bag reconstruction. The total training loss of our method is:

$$L = \sum_i (L_{\text{cls}} + \lambda_{\text{reg}} \mathbb{1}(B_i) L_{\text{reg}} + \lambda_{\text{g}} L_{\text{g}}), \tag{7}$$

Table 1: Comparison with state-of-art methods on THUMOS14 test set under four boundary noise levels. The average mAPs are computed under the IoU thresholds [0.3:0.1:0.7]. Best results are in bold.

Model	Method	10%			20%			30%			40%		
		0.5	0.7	Avg.	0.5	0.7	Avg.	0.5	0.7	Avg.	0.5	0.7	Avg.
Noisy Model	ReAct[20]	54.9	31.4	52.6	53.5	27.6	50.8	51.6	24.1	48.7	47.4	17.8	44.1
	TemporalMaxer[24]	69.7	41.8	65.7	67.9	39.2	64.2	65.4	32.0	61.0	61.5	25.5	56.7
	ActionFormer[31]	68.2	41.3	65.1	68.1	36.3	63.4	65.1	29.6	59.5	59.6	23.0	54.5
	+PbGaR	69.2	42.6	**65.5**	68.7	41.4	**65.2**	67.6	37.4	**63.9**	64.6	31.9	**60.7**
	TriDet[19]	71.2	43.5	66.8	68.6	39.3	64.5	64.4	29.9	59.7	60.5	23.1	55.2
	+PbGaR	71.1	42.6	**66.9**	69.3	40.6	**65.9**	67.9	37.5	**63.7**	65.5	32.6	**60.7**
Clean Model	ActionFormer[31]	70.9	43.9	66.8	70.9	43.9	66.8	70.9	43.9	66.8	70.9	43.9	66.8
	TriDet[19]	72.7	46.5	68.5	72.7	46.5	68.5	72.7	46.5	68.5	72.7	46.5	68.5

Table 2: Comparison with state-of-art methods on ActivityNet1.3 test set under two boundary noise levels. The average mAPs are computed under the IoU thresholds [0.5:0.05:0.95]. Best results are in bold.

Model	Method	20% Boundary Noise level				40% Boundary Noise Level			
		0.5	0.75	0.95	Avg.	0.5	0.75	0.95	Avg.
Noisy Model	ActionFormer[31]	53.96	36.46	5.25	35.20	51.98	32.25	5.73	32.11
	+PbGaR	54.15	36.86	5.68	**35.22**	54.17	35.35	3.09	**34.13**
	TriDet[19]	54.18	37.14	5.64	35.49	51.80	31.86	4.33	31.49
	+PbGaR	54.31	37.18	5.60	**35.67**	53.98	35.49	2.45	**34.10**
Clean Model	ActionFormer[31]	54.79	37.79	8.31	36.61	54.79	37.79	8.31	36.61
	TriDet[19]	54.89	38.08	8.34	36.92	54.89	38.08	8.34	36.92

Specially, the loss function has three terms. L_{cls} is for instance classification, we adopt focal loss[13] to train it. L_{reg} is a DIOU loss[32] for boundary regression. $\mathbb{1}(B_i)$ is an indicator function that denotes whether a bag B_i is positive or not. L_{g} is for training the MIL based pseudo-boundary generation process, which is given in Eq.5. λ_{reg} and λ_{g} are both balance coefficients.

4 EXPERIMENTS

4.1 Settings

Datasets. Since modern temporal action localization datasets are delicately annotated and contain few inaccurate boundary annotations. To evaluate the performance of our proposed PbGaR method, we simulate noisy boundaries by perturbing the clean ones on two on two common used datasets, THUMOS14[9] and ActivityNet1.3[1]. THUMOS14 is comprised of 412 videos with 200 for training and 212 for validation, including 20 action categories. ActivityNet 1.3 contains 20,000 videos covering 200 action categories. It is divided into three subsets, 50% is training set, 25% is validation set and the rest is test set.

Following [14], we simulate noisy action boundaries by perturbing clean ones. Specially, let (cx, l) denote the center x and duration of an action. We randomly shift and scale an action boundary as follows:

$$\begin{cases} \widehat{cx} = cx + \Delta_x \cdot l, \\ \widehat{l} = (1 + \Delta_l) \cdot l, \end{cases} \tag{8}$$

where Δ_x and Δ_l follow the uniform distribution $U(-r, r)$, r refers to the boundary noisy level. We simulate boundary noise levels varying from 10% to 40% and perform Eq.8 on every action boundary in the training data.

Implementation Details. Our method PbGaR is implemented on ActionFormer[31] and TriDet[19] which are two latest state-of-the-art TAL models. Pre-trained I3D[2] is used as backbone. PbGaR is applied after 5 training epochs. We empirically set α to 0.5 and β to 0.75 in Eq.3. The number of bag reconstruction N in refinement module is set to 2. The loss weight λ_{reg} and λ_{g} are selected from $\{0.01, 0.1, 1\}$ depending on datasets and nosiy level. The memory bank is activated after 11 epochs and γ is set to 0.2 in Eq.6. The rest settings are kept unchanged.

Evaluation Metrics. We evaluate our method using the standard TAL metric, i.e. the mean Average Precision(mAP) at differnet temporal intersection over union (tIoU) thresholds for all datastes. Mean average precision (mAP) measures the average precision across all action categories for a given temporal intersection over union (tIoU) threshold. We also report average mAP over several tIoU thresholds.

4.2 Main Results

We compare our method with several state-of-the-art approaches[19, 20, 24, 31] on THUMOS14[9] and ActivityNet1.3[1]. We denote Clean-Model and Noisy-Model as models trained under clean and noisy training data with the default setting. Our intention here is to validate that our method is robust to noisy data and significantly mitigates the performance degradation of TAL models encountered with inaccurate training data.

Results on the THUMOS14 dataset. Table1 shows the comparison results on the THUMOS14 test set. For the existing representative models[19, 20, 24, 31] listed, we observe that inaccurate boundary annotations significantly deteriorate the detection performance of the vanilla model. Our approach, in contrast, demonstrates greater

Table 3: Analysis of the effectiveness of two main components. Experiments are conducted on THUMOS14 dataset.

Method	PBG	PBR	Noise Level			
			10%	20%	30%	40%
1			65.09	63.43	59.47	54.54
2	✓		65.34	63.54	62.01	58.47
3	✓	✓	**65.52**	**65.22**	**63.87**	**60.67**

robustness to inaccurate action boundaries and alleviates degraded performance by a significant margin especially under high noise levels. For example, under 30% noise level, our method on ActionFormer can boost the detection performance of 4.4%. With an mAP of 65.5% (+5.0%) at tIoU=0.5 and an mAP of 32.6% (+9.5%) at tIoU=0.7, our method markedly boost the performance of TriDet under 40% noise level. The results indicate that the pseudo-boundaries generated and refined by our method can provide more precise supervision signals for model training.

Results on the ActivityNet1.3 dataset. The comparison results on ActivityNet1.3 dataset are reported in Table2. Our approach achieves considerable improvements over the vanilla model. For example, under 40% noise, the ActionFormer suffers from obvious performance drop, e.g., drops from 36.6% to 32.11% under 40% noise. With our PbGaR, it achieve a 2.02% improvement in performance. Even in the case of low noise level, our method is still effective. For example, it enhances the accuracy of the ActionFormer at tIoU=0.95 under 20% noise. Our method also assists TriDet in alleviating a 0.18% performance decline. However, a contradiction arises at different noise levels. At 20% level, our method maintains a stable performance at tIoU=0.95, while drops at 40% level. We attribute it to our method makes a tradeoff between mAP at specific tIoU and average mAP. It focuses more on those segments where the boundary annotation is absolutely wrong. For instance, the improvement is obvious at tIoU=0.5 and tIoU=0.75 under 40%.

4.3 Ablation Study

We conduct ablation experiments on THUMOS14 dataset to validate the effectiveness of two modules in our method and we also analyse parameter sensitivity and the two strategies of PBR in this part.

Analysis on main components . To investigate the effectiveness of the two components of PbGaR, we start from the vanilla ActionFormer on noisy data and then gradually add the two modules of our method on it. The results are shown in Table 3. The first row is the vanilla ActionFormer trained under different boundary noise levels. As we gradually add PBG and PBR module into training, it is evident that both modules boost the performance under different noisy levels of training data. For the PBG module, training under our MIL formulation improve the mAP performance of ActionFormer across various boundary noise levels. For instance, the PBG module achieves 2.54% and 3.93% improvements under 30% and 40% box noise level. The second module PBR further enhances the quality of pseudo boundaries, especially under high noise levels. We observe that the impact of PBR is minor under low boundary noise levels. This is likely attributed to the relatively high quality of the action instances in bags when the noise level is low. The results demonstrate that both modules contribute greatly to our method.

Table 4: Ablation on the starting epoch E of PBG. Experiments are conducted on THUMOS14 dataset under 30% noise.

E	0.3	0.5	0.7	Avg.
1	81.28	65.42	33.18	61.61
6	81.73	65.76	34.00	**62.01**
10	81.16	66.46	32.82	61.65
20	80.70	66.09	31.84	61.41

Table 5: Analysis on two strategies of PBR. Experiments are conducted on THUMOS14 dataset under 30% noise.

BR	MB	0.3	0.5	0.7	Avg.
		81.73	65.76	34.00	62.01
✓		81.08	67.39	**37.77**	63.39
	✓	81.10	66.36	34.52	62.22
✓	✓	**82.05**	**67.60**	37.39	**63.87**

Ablation on the starting epoch of PBG module. The starting epoch of our first module determines when to generate pseudo boundaries and thus affects the quality of pseudo boundaries. We train 35 epochs (containing warmup 5 epochs) ActionFormer with PBG on THUMOS14 dataset under 30% noise. We present results for the choice of the starting epoch E of PBG in Tab4. We observe our PBG module can produce stable improvement and the optimal value is obtained at 6.

Analysis on two strategies of PBR. We validate the effectiveness of the two strategies in pseudo-boundary refinement module: bag reconstruction (BR) and memory bank (MB). To verify the effectiveness of these two strategies, we add the PBG module and use only one refinement strategy from PBR to ActionFormer. Experiments are conducted on THUMOS14 under 30% noise level. As shown in Table5, the first row is the result that we add our first PBG module to ActionFormer. The remain three rows demonstrate that either bag reconstruction or memory bank can benefit the performance of TAL models trained under noisy data. This demonstrates that both strategies improve the quality of the generated pseudo boundaries from first module and the combination of them is a preferred option that can better utilize the capabilities of the TAL model.

5 CONCLUSION

In this paper, we focus on learning with inaccurate boundaries in Temporal Action Localization task. By using classification as a signal, we propose a PbGaR method to deal with the performance degradation of TAL models under noisy boundary annotations. The PbGaR firstly generates more accurate pseudo boundaries for training models and then improve the quality of pseudo boundaries via our refnement module. Extensive experiments on two benchmarks demonstrate that PbGaR effectively cooperate with modern TAL detectors and obtain promising performance with inaccurate action boundary annotations.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*. 961–970.

[2] 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.

[3] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence* 89, 1-2 (1997), 31–71.

[4] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. 2017. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing* 26, 6 (2017), 2825–2838.

[5] Junyu Gao and Changsheng Xu. 2021. Fast video moment retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1523–1532.

[6] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* 31 (2018).

[7] Bo He, Xitong Yang, Le Kang, Zhiyu Cheng, Xin Zhou, and Abhinav Shrivastava. 2022. Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13925–13935.

[8] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. 2019. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2888–2897.

[9] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. 2014. THUMOS challenge: Action recognition with a large number of classes.

[10] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. 2016. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer, 350–365.

[11] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. 2021. Weakly-supervised temporal action localization by uncertainty modeling. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 1854–1862.

[12] Junnan Li, Caiming Xiong, Richard Socher, and Steven Hoi. 2020. Towards noise-resistant object detection with noisy annotations. *arXiv preprint arXiv:2003.01285* (2020).

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[14] Chengxin Liu, Kewei Wang, Hao Lu, Zhiguo Cao, and Ziming Zhang. 2022. Robust Object Detection with Inaccurate Bounding Boxes. In *European Conference on Computer Vision*. Springer, 53–69.

[15] Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International conference on machine learning*. PMLR, 6226–6236.

[16] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*. PMLR, 3355–3364.

[17] Huan Ren, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. 2023. Proposal-Based Multiple Instance Learning for Weakly-Supervised Temporal Action Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2394–2404.

[18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*. PMLR, 4334–4343.

[19] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18857–18866.

[20] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. 2022. React: Temporal action detection with relational queries. In *European conference on computer vision*. Springer, 105–121.

[21] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems* 32 (2019).

[22] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*. PMLR, 5907–5915.

[23] Waqas Sultani, Chen Chen, and Mubarak Shah. 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6479–6488.

[24] Tuan N Tang, Kwonyoung Kim, and Kwanghoon Sohn. 2023. TemporalMaxer: Maximize Temporal Context with only Max Pooling for Temporal Action Localization. *arXiv preprint arXiv:2303.09055* (2023).

[25] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. 2019. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11244–11253.

[26] 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).

[27] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. 2017. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 4325–4334.

[28] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13726–13735.

[29] Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. 2021. Learning to purify noisy labels via meta soft label corrector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 10388–10396.

[30] Youjiang Xu, Linchao Zhu, Yi Yang, and Fei Wu. 2021. Training robust object detectors from noisy category labels and imprecise bounding boxes. *IEEE Transactions on Image Processing* 30 (2021), 5782–5792.

[31] Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*. Springer, 492–510.

[32] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 12993–13000.

[33] Zhaowei Zhu, Tongliang Liu, and Yang Liu. 2021. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10113–10123.