CoLA: Weakly-Supervised Temporal Action Localization with Snippet Contrastive Learning

Can Zhang\textsuperscript{1}, Meng Cao\textsuperscript{1}, Dongming Yang\textsuperscript{1}, Jie Chen\textsuperscript{1,2}, Yuexian Zou\textsuperscript{1,2}  
\textsuperscript{1}School of Electronic and Computer Engineering, Peking University \textsuperscript{2}Peng Cheng Laboratory  
{\{zhangcan, mengcao, yangdongming, zouyx\}@pku.edu.cn; chenj@pcl.ac.cn}

Abstract

Weakly-supervised temporal action localization (WS-TAL) aims to localize actions in untrimmed videos with only video-level labels. Most existing models follow the “localization by classification” procedure: locate temporal regions contributing most to the video-level classification. Generally, they process each snippet (or frame) individually and thus overlook the fruitful temporal context relation. Here arises the single snippet cheating issue: “hard” snippets are too vague to be classified. In this paper, we argue that learning by comparing helps identify these hard snippets and we propose to utilize snippet Contrastive learning to Localize Actions, CoLA for short. Specifically, we propose a Snippet Contrast (SniCo) Loss to refine the hard snippets in feature space, which guides the network to perceive precise temporal boundaries and avoid the temporal interval interruption. Besides, since it is infeasible to access frame-level annotations, we introduce a Hard Snippet Mining algorithm to locate the potential hard snippets. Substantial analyses verify that this mining strategy efficaciously captures the hard snippets and SniCo Loss leads to more informative feature representation. Extensive experiments show that CoLA achieves state-of-the-art results on THUMOS’14 and ActivityNet v1.2 datasets.

1. Introduction

Temporal action localization (TAL) aims at finding and classifying action intervals in untrimmed videos. It has been extensively studied in both industry and academia, due to its wide applications in surveillance analysis, video summarization and retrieval [38, 15, 23], etc. Traditionally, fully-supervised TAL is labor-demanding in its manual labeling procedure, thus weakly-supervised TAL (WS-TAL) which only needs video-level labels has gain popularity.

Most existing WS-TAL methods [39, 27, 30, 26, 14] employ the common attention mechanism or multiple instance learning formulation. Specifically, each input video is divided into multiple fixed-size non-overlapping snippets and the snippet-wise classifications are performed over time to generate the Temporal Class Activation Map/Sequence (T-CAM/T-CAS)[27, 34]. The final localization results are generated by thresholding and merging the class activations. For illustration, we consider the naïve case where the whole process is optimized with a single video-level classification loss and we treat this pipeline as baseline in our paper.

In absence of frame-wise labels, WS-TAL suffers from the single snippet cheating issue: indistinguishable snippets are easily misclassified and hurt the localization performance. To illustrate it, we take CliffDiving in Figure 1 as an example. When evaluated individually, two selected snippets (#2, #3) seem ambiguous and are misclassified: 1) the #2 snippet is incorrectly categorized, thus breaking the time intervals; 2) the #3 snippet is misidentified as an action in baseline, resulting in inaccurately extended action interval boundaries. How to address the single snippet cheating issue? Let’s revisit the case in Figure 1. By comparing
snippets of interest with those “easy snippets” which can be classified effortlessly, action and background can be distinguished more easily. For example, the #2 snippet and the #1 easy action snippet are two different views of a man falling-down process in “CliffDiving”. The #3 snippet is similar to the #4 easy background snippet and can be easily classified as the background class. In light of this, we contend that localizing actions by contextually comparing offers a powerful inductive bias that helps distinguish hard snippets. Based on the above analysis, we propose an alternative, rather intuitive way to address the single snippet cheating issue – by conducting Contrastive learning on hard snippets to Localize Actions, CoLA for short. To this end, we introduce a new Snippet Contrast (SniCo) Loss to refine the feature representations of hard snippets under the guidance of those more discriminative easy snippets. Here these “cheating” snippets are named hard snippets due to their ambiguity.

This solution, however, faces one crucial challenge on how to identify reasonable snippets under our weakly-supervised setting. The selection of hard snippets is non-trivial as there is no specific attention distribution pattern for them. For example, in Figure 1 baseline, #3 hard snippet has a high response value while #2 remains low. Noticing that ambiguous hard snippets are commonly found around boundary areas of the action instances, we propose a boundary-aware Hard Snippet Mining algorithm – a simple yet effective importance sampling technique. Specifically, we first threshold T-CAS and then employ dilation and erosion operations temporally to mine the potential hard snippets. Since the hard snippets may either be action or background, we opt to distinguish them by their relative position. For easy snippets, they locate in the most discriminative parts, so snippets with top-k/bottom-k T-CAS scores are selected as easy action/background respectively. Moreover, we form two hard-easy contrastive pairs and conduct the feature refinement via the proposed SniCo Loss.

In a nutshell, the main contributions of this work are as follows: (1) Pioneeringly, we introduce the contrastive representation learning paradigm to WS-TAL and propose a SniCo Loss which effectively refines the feature representation of hard snippets. (2) A Hard Snippet Mining algorithm is proposed to locate potential hard snippets around boundaries, which serves as an efficient sampling strategy under our weakly-supervised setting. (3) Extensive experiments on THUMOS’14 and ActivityNet v1.2 datasets demonstrate the effectiveness of our proposed CoLA.

2. Related Work

Fully-supervised Action Localization utilizes frame-level annotations to locate and classify the temporal intervals of action instances from long untrimmed videos. Most existing works may be classified into two categories: proposal-based (top-down) and frame-based methods (bottom-up). Proposal-based methods [35, 47, 40, 7, 5, 33, 19, 17, 44, 16] first generate action proposals and then classify them as well as conduct temporal boundary regression. On the contrary, frame-based methods [18, 2, 22, 46] directly predict frame-level action category and location followed by some post-processing techniques.

Weakly-Supervised Action Localization only requires video-level annotations and has drawn extensive attention. UntrimmedNets [39] address this problem by conducting the clip proposal classification first and then select relevant segments in a soft or hard manner. STPN [27] imposes a sparsity constraint to enforce the sparsity of the selected segments. Hide-and-seek [36] and MAAN [43] try to extend the discriminative regions via randomly hiding patches or suppressing the dominant response, respectively. Zhong et al. [48] introduce a progressive generation procedure to achieve similar ends. W-TALC [30] applies the deep metric learning to be complementary with the Multiple Instance Learning formulation.

Discussion. The single snippet cheating problem has not been fully studied though it is common in WS-TAL. Liu et al. [20] pinpoint the action completeness modeling problem and the action-context separation problem. They develop a parallel multi-branch classification architecture with the help of the generated hard negative data. In contrast, our CoLA unifies these two problems and settles them in a lighter way with the proposed SniCo Loss. DGAM [32] mentions the action-context confusion issue, i.e., context snippets near action snippets tend to be misclassified, which can be considered as a sub-problem of our single snippet cheating issue. Besides, several background modeling works [28, 14, 32] can also be seen as one solution to this problem. Nguyen et al. [28] utilizes an attention mechanism to model both foreground and background frame appearances and guide the generation of the class activation map. BaS-Net [14] introduces an auxiliary class for background and applies an asymmetrical training strategy to suppress the background snippet activation. However, these methods have inherent drawbacks as background snippets are not necessarily motionless and it is difficult to include them into one specific class. By contrast, our CoLA is a more adaptive and explainable solution to tackle these issues.

Contrastive Representation Learning uses data internal patterns to learn an embedding space where associated signals are brought together while unassociated ones are distinguished via Noise Contrastive Estimation (NCE) [8]. CMC [37] presents a contrastive learning framework that maximize mutual information between different views of the same scene to achieve a view-invariant representation. SimCLR [6] selects the negative samples by using augmented views of other items in a minibatch. MoCo [9] uses a momentum updated memory bank of old negative repre-
3. Method

Generally, CoLA (shown in Figure 2) follows the feature extraction (Section 3.1), actionness modeling (Section 3.2) and hard & easy snippet mining (Section 3.3) pipeline. The optimization loss terms and the inference process are detailed in Section 3.4 and Section 3.5, respectively.

3.1. Feature Extraction and Embedding

Assume that we are given a set of \( N \) untrimmed videos \( \{V_n\}_{n=1}^{N} \) and their video-level labels \( \{y_n\}_{n=1}^{N} \), where \( y_n \in \mathbb{R}^C \) is a multi-hot vector, and \( C \) is the number of action categories. Following the common practice [27, 28, 14], for each input untrimmed video \( V_n \), we divide it into multi-frame non-overlapping \( L_n \) snippets, i.e., \( V_n = \{S_{n,i}\}_{i=1}^{L_n} \). A fixed number of \( T \) snippets \( \{S_{n,t}\}_{t=1}^{T} \) are sampled due to the variation of video length. Then the RGB features \( X_n^R = \{x_n^{R,t}\}_{t=1}^{T} \) and optical flow features \( X_n^O = \{x_n^{O,t}\}_{t=1}^{T} \) are extracted with pre-trained feature extractor (e.g., I3D [4]), respectively. Here, \( x_n^{R} \in \mathbb{R}^d \) and \( x_n^{O} \in \mathbb{R}^d \), \( d \) is the feature dimension of each snippet. Afterwards, we apply an embedding function \( f_{\text{embed}} \) over the concatenation of \( X_n^R \) and \( X_n^O \) to obtain our extracted features \( X_n^E = \{x_n^{E,t}\}_{t=1}^{T \times 2d} \). \( f_{\text{embed}} \) is implemented with a temporal convolution followed by the ReLU activation function.

3.2. Actionness Modeling

We introduce the concept Actionness referring to the likelihood of containing a general action instance for each snippet. Before we specify the Actionness Modeling process, let’s revisit the commonly adopted Temporal Class Activation Sequence (T-CAS).

Given the embedded features \( X_n^E \), we apply a classifier \( f_{ds} \) to obtain snippet-level T-CAS. Specifically, the classifier contains a temporal convolution followed by ReLU activation and Dropout. This can be formulated as follows for a video \( V_n \):

\[
A_n = f_{ds}(X_n^E; \phi_{ds}),
\]

where \( \phi_{ds} \) represents the learnable parameters. The obtained \( A_n \in \mathbb{R}^{T \times C} \) represents the action classification results occurring at each temporal snippets.

Then, when it comes to modeling the actionness, one common way is to conduct the binary classification on each snippet, which yet will inevitably bring in extra overheads. Since the generated T-CAS \( A_n \in \mathbb{R}^{T \times C} \) in Eqn. 1 already contains snippet-level class-specific predictions, we simply sum T-CAS along the channel dimension (\( f_{\text{sum}} \)) followed by the Sigmoid function to obtain a class-agnostic aggregation and use it to represent the actionness \( A_n^{\text{ness}} \in \mathbb{R}^T \):

\[
A_n^{\text{ness}} = \text{Sigmoid}(f_{\text{sum}}(A_n)).
\]

3.3. Hard & Easy Snippet Mining

Recall that our aim is to use the easily spotted snippets as a priori to disambiguate controversial snippets. We systematically study the contrastive pair construction process for both hard and easy snippets.

Figure 2. Illustration of the proposed CoLA, which consists of four parts: (a) Feature Extraction and Embedding to obtain the embedded feature \( X_n^E \); (b) Actionness Modeling to gather class-agnostic action likelihood \( A_n^{\text{ness}} \); (c) Hard & Easy Snippet Mining to select hard and easy snippets. (d) Network Training driven by Action Loss and Snippet Contrast (SniCo) Loss.
3.3.1 Hard Snippet Mining

Intuitively, for most snippets located inside the action or background intervals, they are far from the temporal borders with less noise interference and have the relatively trustworthy feature representation. For boundary-adjacent snippets, however, they are less reliable because they are in the transitional areas between action and background, thus leading to ambiguous detection.

Base on the above observations, we argue that boundary-adjacent snippets can serve as the potential hard snippets under the weak supervision setting. Therefore, we build a novel Hard Snippet Mining algorithm to exploit hard snippets from the border areas. Then these mined hard snippets are divided into hard action and hard background according to their locations.

Firstly, we threshold the actionness scores to generate a binary sequence (1 or 0 indicates the action or background location, respectively):

$$A_n^{bin} = \varepsilon(A_n^{ness} - \theta_b),$$

where $\varepsilon(\cdot)$ is the Heaviside step function and $\theta_b$ is the threshold value, i.e., $A_n^{bin}$ is 1 if $A_n^{ness} \geq \theta_b$, 0 otherwise. Then, as shown in Figure 3, we apply two cascaded dilation or erosion operations to expand or narrow the temporal extent of action intervals. The differential areas with the diverse dilation or erosion degree are defined as the hard background or hard action regions:

$$R_n^{inner} = (A_n^{bin}; m)^- - (A_n^{bin}; M)^-,$$

$$R_n^{outer} = (A_n^{bin}; M)^+ - (A_n^{bin}; m)^+,$$

where $(\cdot; :)^+$ and $(\cdot; :)^-$ represent the binary dilation and erosion operations with mask $\cdot$, respectively. The inner region $R_n^{inner}$ is defined as the different snippets between the eroded sequences with smaller mask $m$ and larger mask $M$, as shown in Figure 3 left part in (green color). Similarly, the outer region $R_n^{outer}$ is calculated as the difference between the dilated sequences with larger mask $M$ and smaller mask $m$, depicted in Figure 3 right part in (pink color). Empirically, we regard the inner regions $R_n^{inner}$ as hard action snippet sets since these regions are with $A_n^{inner} = 1$. Similarly, the outer regions $R_n^{outer}$ are considered as hard background snippet sets. Then the hard action snippets $X_n^{HA} \in \mathbb{R}^{k_{hard} \times 2d}$ are selected from $R_n^{inner}$:

$$X_n^{HA} = \{X_n|t \in T_n^{act}, t \in T_n^{inner}\},$$

where $T_n^{inner}$ is the index set of snippets within $R_n^{inner}$. $T_n^{act}$ is the subset of $T_n^{inner}$ with size $k_{hard}$, i.e., $|T_n^{act}| = k_{hard}$, and $k_{hard} = \max(1, \left\lfloor \frac{|T_n^{inner}|}{v_{\text{hard}}} \right\rfloor)$ is the hyper-parameter controlling the selected number of hard snippets, $v_{\text{hard}}$ is the sampling ratio. Considering the case that $k_{hard} > |T_n^{inner}|$, we adopt sampling with replacement mechanism to ensure the total $k_{hard}$ snippets can be selected. Similarly, the hard background snippets $X_n^{HB} \in \mathbb{R}^{k_{hard} \times 2d}$ are selected from $R_n^{outer}$:

$$X_n^{HB} = \{X_n|t \in T_n^{bkg}, t \in T_n^{outer}\},$$

where the notation definitions are similar to those in Eqn. 5 and we omit them for brevity.

3.3.2 Easy Snippet Mining

In order to form contrastive pairs, we still need to mine the discriminative easy snippets. Based on the well-trained fully-supervised 3D features, we hypothesize that the video snippets with top-$k$ and bottom-$k$ actionness scores are exactly easy action ($X_n^{EA} \in \mathbb{R}^{k_{easy} \times 2d}$) and easy background snippets ($X_n^{EB} \in \mathbb{R}^{k_{easy} \times 2d}$), respectively. Therefore, we conduct easy snippet mining based on the actionness scores calculated in Eqn. 2. The specific process is as follows:

$$X_n^{EA} = \{X_n|t \in S_n^{act}, t \notin T_n^{inner}, t \notin T_n^{outer}\},$$

$$S_n^{act} = S_n^{DESC}[1: k_{easy}]$$

$$X_n^{EB} = \{X_n|t \in S_n^{bkg}, t \notin T_n^{inner}, t \notin T_n^{outer}\},$$

$$S_n^{bkg} = S_n^{ASC}[1: k_{easy}]$$

where $S_n^{DESC}$ and $S_n^{ASC}$ denotes the index of $A_n^{ness}$ sorting by DESC and ASC order respectively. $k_{easy} = \max(1, \left\lfloor \frac{|T_n^{inner}|}{v_{\text{easy}}} \right\rfloor)$, $v_{\text{easy}}$ is a hyper-parameter representing the selection ratio. Note that we remove the snippets in the hard snippet areas $R_n^{inner}$ and $R_n^{outer}$ to avoid conflict.

3.4. Network Training

Based on the mined hard and easy snippets, our CoLA introduces an additional Snippet Contrast (SniCo) Loss ($\mathcal{L}_s$) and achieves considerable improvement compared with the baseline model. The total loss can be represented as follows:

$$\mathcal{L}_{total} = \mathcal{L}_a + \lambda \mathcal{L}_s,$$

where $\mathcal{L}_a$ and $\mathcal{L}_s$ denote the Action Loss and the SniCo Loss, respectively. $\lambda$ is the balance factor. We elaborate on these two terms as follows.
3.4.1 Action Loss

Action Loss ($\mathcal{L}_a$) is the classification loss between the predicted video category and the ground truth. To get the video-level predictions, we aggregate snippet-level class scores computed in Eqn. 1. Following [39, 30, 14], we take the top-keasy strategy: for each class $c$, we take $k_{easy}$ terms with the largest class-specific T-CAS values and compute their means as $a_{n,c}$, namely the video-level class score for class $c$ of video $V_n$. After obtaining $a_{n,c}$ for all the $C$ classes, we apply a Softmax function on $a_n$ along the class dimension to get the video-level class possibilities $p_n \in \mathbb{R}^C$. Action Loss ($\mathcal{L}_a$) is then calculated in the cross-entropy form:

$$\mathcal{L}_a = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log(p_{n,c}), \quad (9)$$

where $y_{n,c}$ is the normalized ground-truth.

3.4.2 Snippet Contrast (SniCo) Loss

Contrastive learning has been used on image or patch levels [1, 10]. For our application, given the extracted feature embedding $X_n^{E}$, the contrastive learning is applied in the snippet level. We name it Snippet Contrast (SniCo) Loss ($\mathcal{L}_s$), which aims to refine the snippet-level feature of hard snippets and obtain a more informative feature distribution. Considering that the hard snippets are classified as hard action and hard background, we form two contrastive pairs in $\mathcal{L}_s$ accordingly, namely “HA refinement” and “HB refinement”, where HA and HB are short for hard action and hard background respectively. “HA refinement” aims to transform the hard action snippet features by driving hard action and easy action snippets compactly in feature space and “HB refinement” is similar.

Formally, the query $x \in \mathbb{R}^{1 \times 2d}$, positive $x^+ \in \mathbb{R}^{1 \times 2d}$, and $S$ negatives $x^- \in \mathbb{R}^{S \times 2d}$ are selected from pre-mined snippets. As shown in Figure 2(d), for “HA refinement”, $x \sim X_n^{HA}, x^+ \sim X_n^{EA}, x^- \sim X_n^{EB}$, for “HB refinement”, $x \sim X_n^{HB}, x^+ \sim X_n^{EB}, x^- \sim X_n^{EA}$. We project them to a normalized unit sphere to prevent the space from collapsing or expanding. An $(S + 1)$-way classification problem using the cross-entropy loss is set up to represent the probability of the positive example being selected over negatives. Following [9], we compute the distances between the query and other examples with a temperature scale $\tau = 0.07$:

$$\ell(x, x^+, x^-) = -\log \left[ \frac{\exp \left( x^T \cdot x^+ / \tau \right)}{\exp \left( x^T \cdot x^+ / \tau \right) + \sum_{s=1}^{S} \exp \left( x^T \cdot x^s / \tau \right)} \right], \quad (10)$$

where $x^T$ is the transpose of $x$ and the proposed SniCo Loss is as follows:

$$\mathcal{L}_s = \mathbb{E}_{x \sim X_n^{HA}}, x^+ \sim X_n^{EA}, x^- \sim X_n^{EB} \ell(x, x^+, x^-) \quad \text{(HA refinement)}$$

$$\mathbb{E}_{x \sim X_n^{HB}}, x^+ \sim X_n^{EB}, x^- \sim X_n^{EA} \ell(x, x^+, x^-) \quad \text{(HB refinement)} \quad (11)$$

where $S$ represents the number of negative snippets and $x^s \in \mathbb{R}^{2d}$ means the $s$-th negative. In this way, we maximize mutual information between the easy and hard snippets of the same category (action or background), which helps refine the feature representation and thereby alleviating the single snippet cheating issue.

3.5. Inference

Given an input video, we first predict its snippet-level class activations to form T-CAS and aggregate top-keasy scores described in Sec. 3.4.1 to get the video-level predictions. Then the categories with scores larger than $\theta$, are selected for further localization. For each selected category, we threshold its corresponding T-CAS with $\theta$, to obtain candidate video snippets. Finally, continuous snippets are grouped into proposals and Non-Maximum Suppression (NMS) is applied to remove duplicated proposals.

4. Experiments

4.1. Datasets

We evaluate our CoLA on two popular action localization benchmark datasets including THUMOS’14 [11] and ActivityNet v1.2 [3]. We only use the video-level category labels for network training.

THUMOS’14 includes untrimmed videos with 20 categories. The video length varies greatly and each video may contain multiple action instances. By convention [14, 32], we use the 200 videos in validation set for training and the 213 videos in testing set for evaluation.

ActivityNet v1.2 is a popular large-scale benchmark for TAL with 100 categories. Following the common practice [39, 34], we train on the training set with 4,819 videos and test on the validation set with 2,383 videos.

4.2. Implementation Details

Evaluation Metrics. We follow the standard evaluation protocol by reporting mean Average Precision (mAP) values under different intersection over union (IoU) thresholds. The evaluation on both datasets is conducted using the benchmark code provided by ActivityNet1.

1https://github.com/activitynet/ActivityNet/
Feature Extractor. We use I3D [4] network pre-trained on Kinetics [4] for feature extraction. Note that the I3D feature extractor is not fine-tuned for fair comparison. TVL1 [31] algorithm is applied to extract optical flow stream from RGB stream in advance. Each video stream is divided into 16-frame non-overlapping snippets and the snippet-wise RGB and optical flow features are 1024-dimension.

Training Details. The number of sampled snippets \( T \) for THUMOS’14 and ActivityNet v1.2 is set to 750 and 50, respectively. All hyper-parameters are determined by grid search: \( r_{\text{easy}} = 8, r_{\text{hard}} = 32, S = k_{\text{easy}} = \max(1, \lfloor \frac{T}{t_{\text{pros}}} \rfloor) \). We set \( \lambda = 5e - 3 \) in Eqn. 8. \( \theta_b \) in Eqn. 3 is set to 0.5 for both datasets. Dilation and erosion masks \( M \) and \( m \) are set to 6 and 3 in our experiments. We utilize Adam optimizer with a learning rate of \( 1e - 4 \). We train for total 6k epochs with a batch size of 16 for THUMOS’14 and for total 8k epochs with a batch size of 128 for ActivityNet v1.2.

Testing Details. We set \( \theta_b \) to 0.2 and 0.1 for THUMOS’14 and ActivityNet v1.2, respectively. For proposal generation, we use multiple thresholds that \( \theta_s \) is set as \([0:0.25:0.025]\) for THUMOS’14 and \([0:0.15:0.015]\) for ActivityNet v1.2, then Non-Maximum Suppression (NMS) is performed with IoU threshold 0.7.

4.3. Comparison with State-of-the-Arts

We compare our CoLA with the state-of-the-art fully-supervised and weakly-supervised TAL approaches on THUMOS’14 testing set. As shown in Table 1, CoLA achieves the impressive performance, i.e., we consistently outperform previous weakly-supervised methods at all IoU thresholds. Specifically, our method achieves 32.2% mAP@0.5 and 40.9% mAP@AVG, bringing the state-of-the-art to a new level. Notably, even with a much lower level of supervision, our method is even comparable with several fully-supervised methods, following the latest fully-supervised approaches with the least gap.

We also conduct experiments on ActivityNet v1.2 validation set and the comparison results are summarized in Table 2. Again, our method shows significant improvements over state-of-the-art weakly-supervised TAL methods while maintaining competitive compared with other fully-supervised methods. The consistent superior results on both datasets signify the effectiveness of CoLA.

4.4. Ablation Studies

In this section, we conduct multiple ablation studies to provide more insights about our design intuition. By convention [28, 32, 14], all the ablation experiments are performed on the THUMOS’14 testing set.

Q1: How does the proposed SniCo Loss help? To evaluate the effectiveness of our SniCo Loss (\( \mathcal{L}_s \)), we conduct a comparison experiment with only the action loss \( \mathcal{L}_a \) as supervision, namely baseline in Table 3. The statistical results in Table 3 demonstrate that by introducing \( \mathcal{L}_s \), the performance largely gains by 7.5% in mAP@0.5, partially because SniCo Loss effectively guides the network to achieve better feature distribution tailored for WSTAL. To illustrate this, we randomly select 2 videos from THUMOS’14 testing set and calculate the feature embeddings \( X_n^E \) for baseline and CoLA, respectively. These embeddings are then projected to 2-dimensional space using UMAP [24], as shown in Figure 4. Notice that compared with baseline, SniCo Loss helps to separate the action and background snippets more precisely, especially for those ambiguous hard snippets. Overall, the above analyses strongly justify the significance of our proposed SniCo Loss.

Q2: Is it necessary to consider both HA and HB refinements in SniCo Loss? To explore this, we conduct ablated experiments with two variants of SniCo Loss, each of which contains only one type of refinement in Eqn. 11, namely \( \mathcal{L}_n^{HA} \) and \( \mathcal{L}_n^{HB} \), respectively. Table 3 shows that the performance drops dramatically with either kind of refinement removed, suggesting that both refinements contribute to the improved performance.

Q3: Are our mined hard snippets meaningful? How to evaluate the effectiveness of the mined hard snippets is nontrivial. As discussed in Sec. 3.3.1, indistinguishable frames usually exist within or near the action temporal intervals, so we define such temporal areas as error-prone.
The mean RDO (mRDO) of all the regions. Specifically, given a ground-truth action instance with interval $[s, e]$ and duration $d = e - s$, we define its $\delta$-scale error-prone regions as $[s - \delta d/2, e + \delta d/2]$, as illustrated in Figure 5 top part. Then, to evaluate the positional relationship of our mined hard snippets with the error-prone areas, relative distance offset (RDO) is defined as follows: 1) if a mined hard snippet does not fall into any of the error-prone regions, $RDO = 0$, where $D$ is the nearest distance between this snippet and all error-prone regions, and $T$ is the video length; 2) otherwise, $RDO = 0$. As shown in Figure 5 bottom part, the mean RDO values (mRDO) of all the videos are evaluated under different scales $\delta$ at two training snapshots (epoch 0 and epoch 2k). The mRDO consistently drops at all scales $\delta$, indicating that our mined hard snippets
are captured more precisely as the training goes on. Even under the most stringent condition ($\delta = 0.2$), the mRDO is only 3.7%, which suggests that most of our mined hard snippets locate in such error-prone areas and thus contribute to the network training.

**Evaluation on the negative sample size $S$.** Table 4 reports the experimental results evaluated with different negative sample sizes $S$. According to Eqn. 11, negative snippets are randomly chosen from the mined easy snippets, so $S \leq k^{\text{easy}}$. As shown, the mAP value is positively correlated with $S$, indicating that contrastive power increases by adding more negatives. This phenomenon is consistent with many self-supervised contrastive learning works [29, 9, 6] and a recent supervised one [13], which partially verifies the efficacy of our hard and easy snippet mining algorithm for weakly-supervised TAL task.

**Evaluation on the mask size $\mathcal{M}$ and $m$.** We have defined two operation degrees (with larger $\mathcal{M}$ and smaller $m$) for temporal interval erosion and dilation in Eqn. 4. Here we seek to evaluate the effect of different mask sizes. For simplification, we first fix $m = 3$ and vary $\mathcal{M}$ from 4 to 9, then we fix $\mathcal{M} = 6$ and change $m$ from 0 to 5. The results are shown in Table 5. The best result is achieved when setting $\mathcal{M} = 6$ and $m = 3$. Besides, it is quite evident that the performance remains stable across a wide range of $\mathcal{M}$ and $m$, demonstrating the robustness of our proposed Hard Snippet Mining algorithm.

### 4.5. Qualitative Results

We visualize T-CAS results for two actions on THUMOS’14 in Figure 6. Our CoLA has a more informative T-CAS distribution compared to baseline, thus leading to more accurate localization. Figure 6-A depicts a typical case that all the frames in a video share the similar elements, *i.e.*, humans, billiard table and balls. By introducing SniCo Loss, our method can seek the subtle differences between action and hard background, thereby avoiding many false positives produced by single Action Loss (baseline). Figure 6-B demonstrates a “CliffDiving” action observed from different camera views. The baseline method fails to localize the complete interval and outputs short and sparse prediction results. Our method successfully identifies the entire “CliffDiving” action and suppress the false positive detections. We also visualize the mined hard snippet locations (computed at epoch 2k) on the time axis (marked as red pentagram). As expected, these snippets are misclassified in baseline and CoLA refines their representation to achieve better performance. This visualization also helps explain Q3 in Section 4.4. For more visualization results, please refer to our supplementary materials.

### 5. Conclusion

In this paper, we have proposed a novel framework (CoLA) to address the single snippet cheating issue in weakly-supervised action localization. We leverage the intuition that hard snippets frequently lay in the boundary regions of the action instances and propose a Hard Snippet Mining algorithm to localize them. Then we apply a SniCo Loss to refine the feature representation of the mined hard snippets with the help of easy snippets which locate in the most discriminative regions. Experiments conducted on two benchmarks including THUMOS’14 and ActivityNet v1.2 have validated the state-of-the-art performance of CoLA.

**Acknowledgements.** This paper was partially supported by the IER foundation (No. HT-JD-CXY-201904) and Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing). Special acknowledgements are given to Aoto-PKUSZ Joint Lab for its support.
References


[26] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for...


