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

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

In this material, we provide additional ablation studies (§1), experiment details of baseline (§2) and more qualitative visualizations (§3 & §4). Moreover, video demos including one "billiards" action in THUMOS'14 and one video clip from the classic British sitcom "Mr Bean" are attached as separate files. Note that we do not fine-tune our model on extra data.

1. More ablations

To further demonstrate the effectiveness of our proposed Hard & Easy Snippet Mining algorithm, we set up a set of comparison experiments:

1) We drop the Easy Snippet Mining and randomly select easy snippets X_n^{EA} and X_n^{EB} from the embedding feature X_n^E (w/o ESM in Table 1).

2) We drop the Hard Snippet Mining and randomly select hard snippets X_n^{HA} and X_n^{HB} from X_n^E (w/o HSM in Table 1).

From Table 1, we can conclude that dropping either the Hard or Easy Snippet Mining process leads to significant performance degradation. For example, mAP drops by 4.1% when replacing Easy Snippet Mining with random selection. Notably, even only equipped with single snippet mining process, both of the two variants (w/o ESM and w/o HSM) outperform the baseline model, which also demonstrates the effectiveness of our proposed mining algorithm.

Table 1: Effectiveness evaluation of Hard & Easy Snippet Mining algorithm on THUMOS'14.

Setting	Loss	mAP@0.5 (Δ)
CoLA (Ours)	$\mathcal{L}_a + \mathcal{L}_s$	**32.2%**
baseline	$\mathcal{L}_a$	24.7% (-7.5%)
CoLA w/o ESM	$\mathcal{L}_a + \mathcal{L}_s$	28.1% (-4.1%)
CoLA w/o HSM	$\mathcal{L}_a + \mathcal{L}_s$	26.3% (-5.9%)

2. Experiment Details of Baseline

The overall architecture of the *baseline* mentioned in our paper is illustrated in Figure 1. To have an apple-to-apple comparison, the implementation details of baseline are identical to those of our CoLA. During training, it first performs feature preparation and then conducts action classification to get temporal class activation sequences (T-CAS). The whole network is optimized with a single Action Loss, which is detailed in our main paper. Specifically, f_{embed} is implemented with a temporal convolution (kernel_size=3 and filter_size=$2d$, *i.e.*, 2048) followed by the ReLU activation function. The classifier f_{cls} contains a temporal convolution (kernel_size=3 and filter_size=C, where C is the number of categories) followed by ReLU activation and Dropout (ratio=0.7). Same as CoLA, class-level top-k^{easy} mean values are generated from T-CAS to derive video-level class score, where $k^{easy} = \frac{T}{r^{easy}}$ and r^{easy}=8. During testing, the localization results are generated by thresholding and merging the class activations. The Non-maximum Suppression (NMS) threshold is set as 0.7.

3. More Localization Visualizations

We visualize more localization results and the T-CAS for both the baseline and our CoLA. For more intuitive comparisons, we also plot the results of two recent works Liu *et al.* [2] and BaSNet [1]. Concretely, we directly utilize the codebase and models in the official implementations of [2] [1] and [1] [2]. Note that for fair comparison, we do not fine-tune any model parameters and codes provided by the authors and just perform inference. Several qualitative results on THUMOS'14 are shown in Figure 2 (sparse case: sparse action intervals within videos) and Figure 3 (dense case: dense action intervals within videos). Obviously, our CoLA consistently outperforms other listed methods on these challenging cases. The detailed analyses can be found in the captions below each sub-figure.

[1]https://github.com/Finspire13/CMCS-Temporal-Action-Localization
[2]https://github.com/Pilhyeon/BaSNet-pytorch

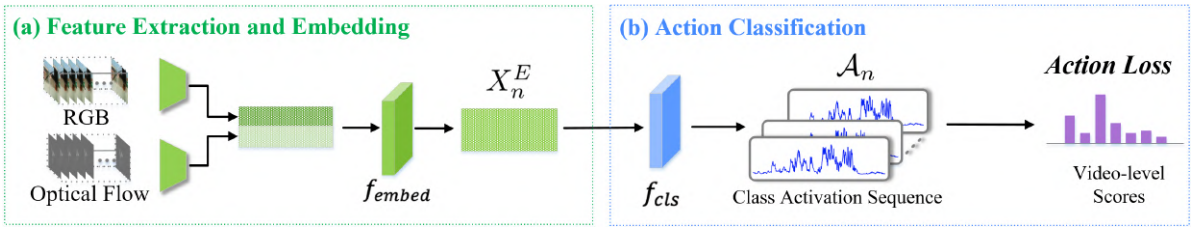


Figure 1: Illustration of the baseline, which consists of two parts: (a) Feature Extraction and Embedding to obtain the embedded feature X_n^E; (b) Action Classification to gather video-level class scores. The whole network is optimized with a single Action Loss.

4. More UMAP Visualizations of Feature Embeddings

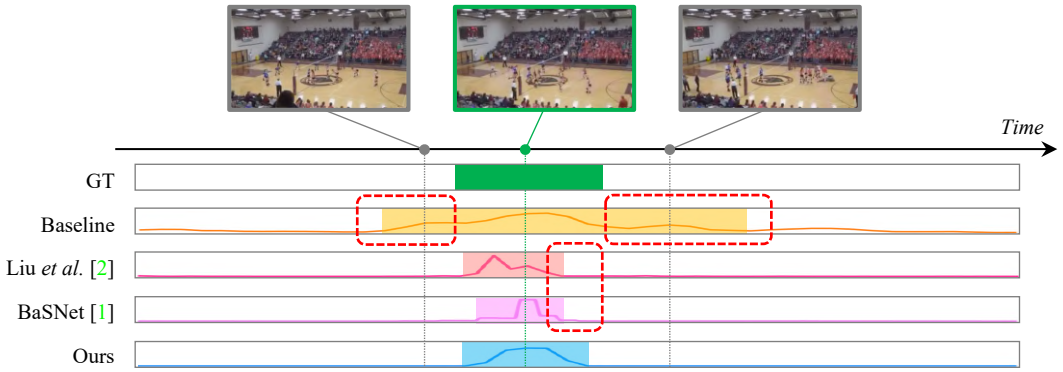
In this section, more visualizations of learned feature embeddings are presented. We first randomly select videos from THUMOS'14 testing set and calculate their feature embeddings X_n^E for baseline and CoLA, respectively. These embeddings are then projected to 2-dimensional space using UMAP [3], as 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.

References

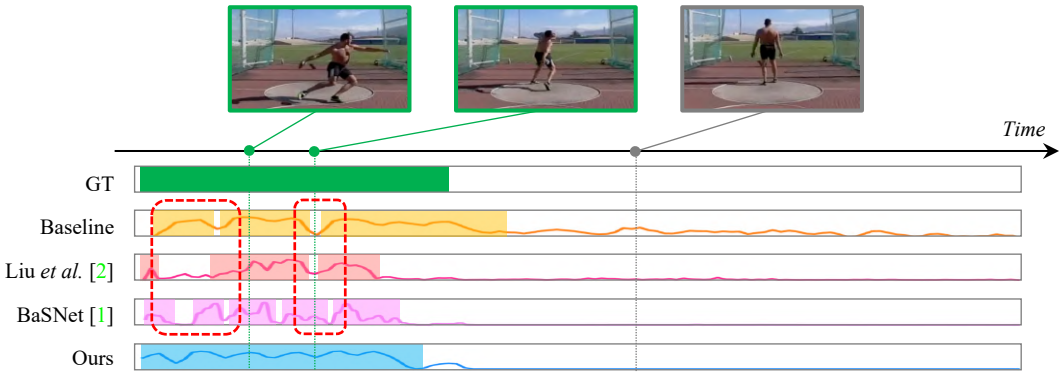
[1] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, pages 11320–11327, 2020. 1

[2] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1298–1307, 2019. 1

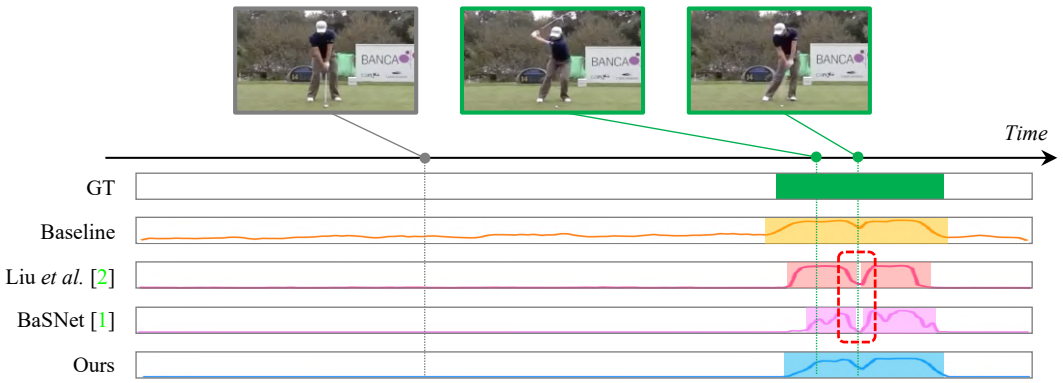
[3] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, Feb. 2018. 2



(a) "*VolleyballSpiking*" action. It is a challenging case that humans look small and actions sparsely occur *i.e.*, background frames dominate the video. Despite these challenges, our method successfully suppresses the activations of background frames and further achieves more complete and precise localization results than others.

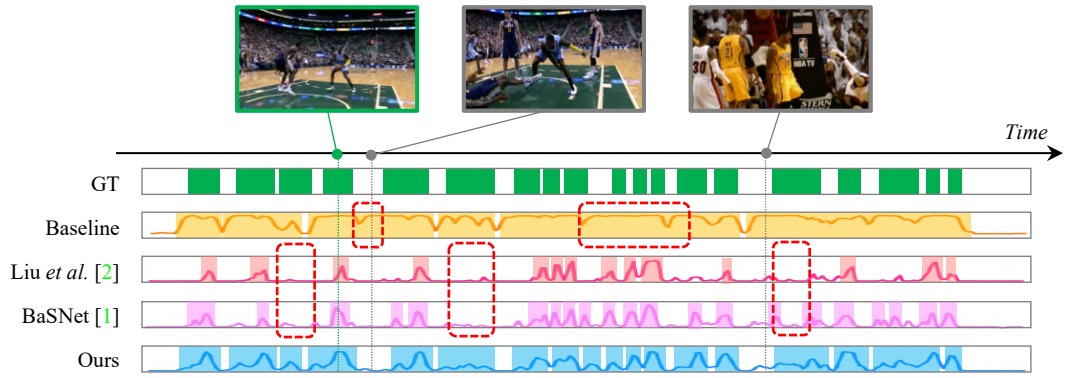


(b) "*ThrowDiscus*" action. All the frames in this video share the similar elements, *i.e.*, athlete, track field and a throwing circle. Our CoLA outputs more smooth and continuous T-CAS, partially because temporal context relation can be linked through contrastive learning, which will mitigate the "false negative" problem.

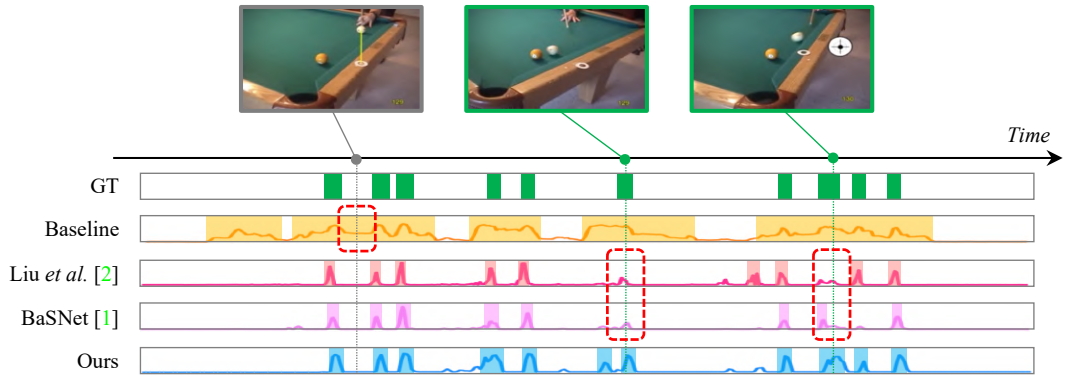


(c) "*GolfSwing*" action. All the frames in this video have similar appearances. Notably, the depicted 3rd frame (action, downswing) is very similar to the 1st frame (background, pre-swing). Our method successfully distinguish those "hard snippets" and further seeks the action interval more precisely.

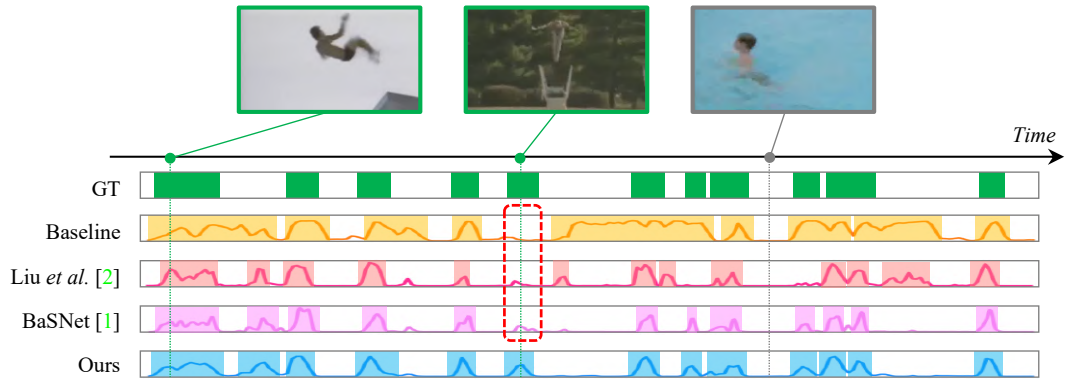
Figure 2: More Localization Visualizations (**Sparse Case**). For baseline and CoLA, we visualize the one-dimensional T-CAS and the localized regions. For clarity, frames with green bounding boxes refer to ground-truth actions and those in gray refer to ground-truth backgrounds.



(a) "*BasketballDunk*" action. The video has significantly dense action instances, making the localization difficult. The baseline suffers from severe "false positive" problem, while Liu *et al.* and BaSNet produce many "false negative" localization results. In contrast, by contrastively refining hard snippets under the guidance of easy snippets, our method can accurately pinpoint the target actions.

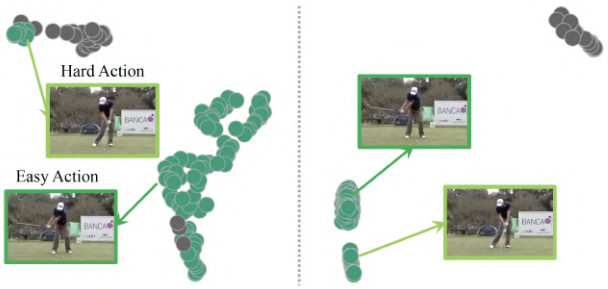


(b) "*Billiards*" action. In this case, stationary commentary clips (1st frame) are semantically similar to the true actions (2nd and 3rd frames). The baseline fails to distinguish them, while our CoLA effectively filters out these error-prone clips with our contrastive idea.

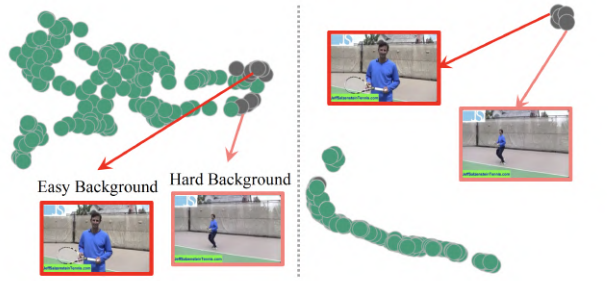


(c) "*Diving*" action. It is a quite challenging case since the viewpoint of the camera changes widely, making the scenes of the same action instance look very different. All the other methods fail to locate the 2nd frame, which lead to incomplete results. Surprisingly, our method successfully pinpoints the complete "Diving" action, which proves the effectiveness of our proposed CoLA.

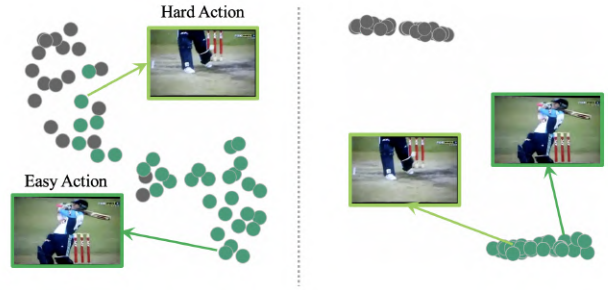
Figure 3: More Localization Visualizations (**Dense Case**). For baseline and CoLA, we visualize the one-dimensional T-CAS and the localized regions. For clarity, frames with green bounding boxes refer to ground-truth actions and those in gray refer to ground-truth backgrounds.



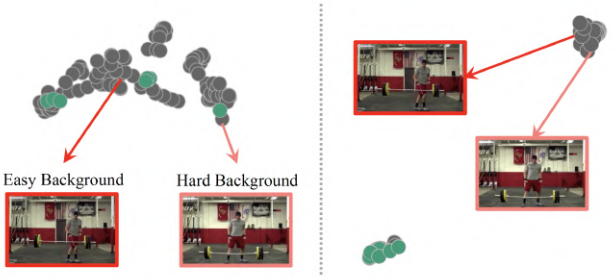
(a) "*GolfSwing*" action



(b) "*TennisSwing*" action



(c) "*CricketShot*" action



(d) "*CleanAndJerk*" action

Figure 4: UMAP visualizations of feature embedding X_n^E. Left: baseline; Right: CoLA. Green points represent action embeddings and gray points denote background embeddings. CoLA achieves a more separable feature distribution compared to baseline.