Proposal-based Temporal Action Localization with Point-level Supervision Supplementary Material

Yuan Yin
yinyuan@iis.u-tokyo.ac.jp
Yifei Huang
hyf@iis.u-tokyo.ac.jp
Ryosuke Furuta
furuta@iis.u-tokyo.ac.jp
Yoichi Sato
ysato@iis.u-tokyo.ac.jp

Institute of Industrial Science
The University of Tokyo
Tokyo, JP

In this supplementary material, we first discuss the pseudo label generation algorithm in detail. Then we present ablation studies to prove the effectiveness of vision-language modeling and our method's robustness to point-level annotations in different settings. Finally, we show some qualitative results and analyze the limitations and promising directions to improve our method.

1 Pseudo Label Generation Algorithm

In this section, we will explain our pseudo label generation algorithm in detail. We first perform constrained k-medoids clustering to find the action boundaries based on point-level annotations. Then we mine background frames in order to provide more fine-grained supervision. Following this process, we will first demonstrate the constrained k-medoids clustering and then introduce the background mining.

As discussed in the main submission, in order to output the temporal boundaries of each action instance and guarantee each action instance is consistent temporally, we find the temporal location (boundary) that divides the frames between two consecutive point-level annotations into two clusters (action instances) where the distance between the boundary and the cluster medoid is minimized. Formally, given embedded features $\{x_i\}_{i=1}^{T}$ and point-level annotations $\{t_i\}_{i=1}^{N}, t_i < t_{i+1}$ of the input video, we aim to output the action boundaries $\{b_i\}_{i=0}^{N}$, where $t_{i-1} < b_i < t_i$. A simple clustering algorithm to address this problem is to find the boundary b_i that:

$$b_i = \arg\min_{p} \left(\sum_{j=t_i}^{p} dist(x_j, m_i) + \sum_{j=p+1}^{t_{i+1}} dist(x_j, m_{i+1}) \right) \tag{1}$$

where $dist(\cdot)$ is the Euclidean distance, $m_i = x_{t_i}$ is the medoid for i-th cluster (action instance). Intuitively, b_i is an estimation of the decision boundary that partitions the frames in

$[t_i, t_{t+1}]$ into two clusters.

However, we argue that this simple approach cannot give a reasonable and robust estimation. This is because in Equation 1, b_i is calculated with "static" cluster medoids m_i, m_{i+1}, which are kept the same during the clustering process. If cluster medoids are initialized with "bad" point-level annotations that are not representative enough for the corresponding clusters, then the resulting b_i will be bad predictions. Inspired by the forward-backward algorithm in [1], we design an efficient k-medoids clustering algorithm that keeps updating the cluster medoids based on previous predictions. The motivation is as follows: if we get the prediction of b_i, then we already know that the frames in $[b_{i-1}, b_i]$ belong to the same i-th action instance (cluster). Therefore, we update m_i as the average of $x_{b_{i-1}:b_i}$ and use the updated medoid to estimate the next action boundary b_{i+1}. We call it a *forward pass* and denote the resulting *forward predictions* as $\{b_i^F\}_{i=0}^N$. Similarly, we can get *backward predictions* $\{b_i^B\}_{i=0}^N$ through a *backward pass*. We reach the final predictions for $\{b_i\}_{i=0}^N$ by averaging forward and backward predictions. Algorithm 1 shows the pseudo-code of our proposed clustering algorithm.

Algorithm 1 Constrained K-medoids Clustering Algorithm

Input: video length T; embedded features $\{x_i\}_{i=1}^T$; point-level annotations $\{t_i\}_{i=1}^N, t_i < t_{i+1}$

Output: action boundaries $\{b_i\}_{i=0}^N$ with the constraint that $b_0 = 0, b_N = T, t_i < b_i < t_{i+1}$; updated cluster medoids $\{m_i\}_{i=1}^N$

Init: cluster medoids $\{m_i\} = \{x_{t_i}\}_{i=1}^N$; action boundaries $b_0^F = b_0^B = 0, b_N^F = b_N^B = T$

repeat

  for $i = 1, ..., N-1$ **do** ▷ calculate forward predictions

    $b_i^F = \arg\min_p \left(\sum_{j=t_i}^{p} dist(x_j, \frac{1}{p - b_{i-1}^F + 1} \sum_{k=b_{i-1}^F}^{p} x_k) + \sum_{j=p+1}^{t_{i+1}} dist(x_j, m_{i+1}) \right)$

  end for

  for $i = N-1, ..., 1$ **do** ▷ calculate backward predictions

    $b_i^B = \arg\min_p \left(\sum_{j=t_i}^{p} dist(x_j, m_i) + \sum_{j=p+1}^{t_{i+1}} dist(x_j, \frac{1}{b_{i+1}^B - p} \sum_{k=p+1}^{b_{i+1}^B} x_k) \right)$

  end for

  for $i = 1, ..., N$ **do**

    $b_i = \frac{1}{2} \left(b_i^B + b_i^F \right)$ ▷ average the forward and backward predictions

    $m_i = \frac{1}{b_i - b_{i-1} + 1} \sum_{j=b_{i-1}}^{b_i} x_j$ ▷ update cluster medoids

  end for

until convergence

return $\{b_i\}_{i=0}^N, \{m_i\}_{i=1}^N$

The proposed clustering algorithm outputs action boundaries regardless of background instances between actions. To generate more accurate pseudo labels, we also propose to mine background frames. We assume that at least one background frame exists between two consecutive action instances to separate them. As $\{b_i\}_{i=0}^N$ are predictions of the moments when the transition between two consecutive action instances happens, they offer helpful hints of background frames. We regard b_i as the anchor frame for i-th background instance and expand from it. Once the distance between the current frame and b_i is larger than a pre-defined threshold, then we denote the current frame as the boundary of i-th background instance.

Technically, given action boundaries $\{b_i\}_{i=0}^{N}$ and updated cluster medoids $\{m_i\}_{i=1}^{N}$, we traverse from b_i to t_i and find the first temporal location δ_i^- such that: $\frac{dist(m_i,\delta_i^-)}{dist(m_i,b_i)} < \zeta$, where ζ is a pre-defined threshold. The resulting δ_i^- is the left (start) boundary of i-th background instance. Similarly, we traverse from b_i to t_{i+1} and find the first temporal location δ_i^- such that: $\frac{dist(m_{i+1},\delta_i^+)}{dist(m_{i+1},b_i)} < \zeta$. The resulting δ_i^+ is the right (end) boundary of i-th background instance. For the corner cases b_0 and b_N, we only calculate δ_0^+ and δ_N^-, and directly let $\delta_0^- = b_0, \delta_N^+ = b_N$. Finally, we assign the frames in (δ_i^-, δ_i^+) as background frames and denote $\varphi_i^b = (\delta_i^-, \delta_i^+)$ as i-th pseudo labeled background instance. Meanwhile, we assign the frames in $[\delta_{i-1}^+, \delta_i^-]$ with the action labels of t_i point-level annotation and denote $\varphi_i = (\delta_{i-1}^+, \delta_i^-, c_i)$ as i-th pseudo labeled action instance.

2 Point-level Annotation Generation

As mentioned in the main submission, all the four benchmarks do not have official (human-annotated) point-level annotations. Therefore, we need to simulate point-level annotations from their available frame-level annotations. For each action instance, we denote the ground truth start and end time as t^s, t^e. Then we randomly sample one frame from the normal distribution with mean $\frac{t^s+t^e}{2}$ and a standard deviation of 1 second. We denote such point-level annotations as "Normal". To verify the robustness of our method to point-level annotations under different distributions, we show the results on ActivityNet 1.3 in two additional distributions. "Uniform" indicates that the point-level annotations are randomly sampled from a uniform distribution $[t^s, t^e]$, and "Center" indicates that we directly use $\frac{t^s+t^e}{2}$ as the point-level annotations. From Table 1, we can see that our model is robust to point-level annotations from different distributions.

Table 1: Comparison of our method's performance on ActivityNet 1.3 trained with point-level annotations from different distributions.

Point-level Annotations	mAP@IoU (%) 0.5	0.75	0.95	AVG
Uniform	46.1	26.5	6.2	27.9
Center	**48.7**	27.5	6.5	28.8
Normal	48.3	**27.8**	**7.0**	**29.1**

3 Vision-language Modeling

To show the effectiveness of vision-language modeling (VLM), we compare two implementations of PEM. In the original implementation of PEM, we use two Transformers [3] as vision encoder and text encoder respectively. In order to exclude the effect of VLM, we remove the text encoder and replace the vision encoder with a larger Transformer so that the number of parameters of PEM is unchanged. We use a linear layer on top of the output of the vision encoder to directly output classification scores and confidence scores for proposals. We train the resulting model in the same experiment setting on ActivityNet 1.3 dataset, and the results are shown in Table 2. We can see that with the help of VLM, the mAPs at

all IoU thresholds improve, which shows that VLM is beneficial for more accurate action localization.

Table 2: Abaltion studies of the PEM with vision-language modeling on ActivityNet 1.3.

Backbone of PEM	mAP@IoU (%)			AVG
	0.5	0.75	0.95	
Transformer w/o VLM	47.5	27.1	6.4	28.1
Transformer w VLM	**48.3**	**27.8**	**7.0**	**29.1**

4 Qualitative Results

Figure 1 shows the qualitative results of our methods and the state-of-the-art MIL-based method [2] on THUMOS 14 dataset. We can observe that our method outputs more precise predictions. Additionally, our proposal-based method greatly addresses the incomplete localization error in the MIL-based method. Specifically, in Figure 1(a), [2] cannot distinguish the starting parts of "*CleanAndJerk*" well. This is because at the starting phase of weightlifting, a man tends to move very slowly. As we discussed in the main submission, these slow motions are difficult for MIL-based method to distinguish. In contrast, with the help of action proposals that contain extensive temporal information, we successfully output more accurate predictions. We also show an example with failure cases in Figure 1(b). Generally, the predictions of our method are still more accurate than the incomplete predictions of [2]. But the third and the last predicted action instances are over-complete due to the short duration and short interval between two consecutive actions.

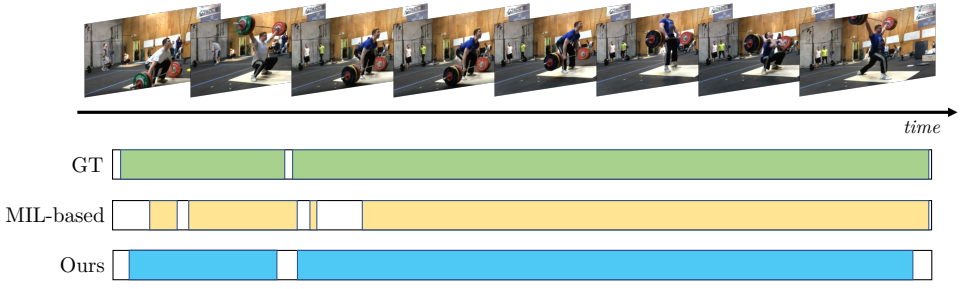
5 Limitations and Future Work

As indicated by the second example of qualitative results, a major factor that may influence the performance of our model is the high frequency and short duration of action instances in the input videos. When there are a large number of short action instances in one video, it will become very difficult for our BDM and PGM to generate action proposals accurately because of the short duration of actions and short intervals between actions, thus leading to performance degradation.

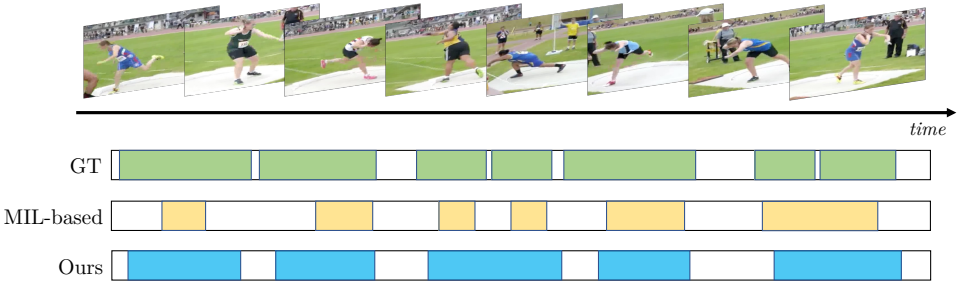
Table 3: Comparison results of our model trained under different levels of supervision on ActivityNet 1.3.

Supervision	mAP@IoU (%)			AVG
	0.5	0.75	0.95	
Frame-level	**52.7**	**35.3**	**8.3**	**33.7**
Point-level	48.3	27.8	7.0	29.1

Another limitation is the quality of the generated pseudo labels used for training because the training of our model largely relies on those pseudo labels. We conduct an additional experiment on ActivityNet 1.3 where our method is trained with frame-level labels (full supervision) and we do not generate pseudo labels. In other words, this is equivalent to what



(a) An example of "*CleanAndJerk*"



(b) An example of "*Shotput*"

Figure 1: Qualitative results on THUMOS 14. We present two examples from "CleanAndJerk" and "Shotput" respectively. In each example, we present the ground truth (top), the final prediction of [2] (mid), and ours (bottom). Our method achieves obviously more complete predictions.

we would get if the generated pseudo labels were totally correct. As shown in Table 3, more accurate labels bring notable improvement in performance, which shows the importance of improving the quality of pseudo labels. In other words, exploring a more effective pseudo label generation algorithm is promising to improve our method, and we leave it for future work.

References

[1] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[2] Pilhyeon Lee and Hyeran Byun. Learning action completeness from points for weakly-supervised temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13648–13657, 2021.

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