ASM-Loc: Action-aware Segment Modeling for Weakly-Supervised Temporal Action Localization
Supplementary Material

Bo He[1], Xitong Yang[1], Le Kang[2], Zhiyu Cheng[2], Xin Zhou[2], Abhinav Shrivastava[1]

[1]University of Maryland, College Park [2]Baidu Research, USA

{bohe,xyang35,abhinav}@cs.umd.edu, {kangle01,zhiyucheng,zhouxin16}@baidu.com

Sec. 1 reports additional experiments and analysis. Sec. 2 elaborates on the procedure of action proposal generation. Sec. 3 provides more dataset-specific implementation details and hyper-parameters for training and testing. We also provide more qualitative results in Sec. 4. We discuss the limitation and broader impact of our work in Sec. 5 and Sec. 6.

1. Additional Experiments and Analysis

Error analysis. To analyze the effectiveness of our ASM-Loc, we conduct a DETAD [1] false positive analysis of the base model without any action-aware segment modeling modules and our ASM-Loc. We present the results in Figure 4. It shows a detailed categorization of false positive errors and summarizes the distribution of these errors. G represents the number of ground truth segments in the THUMOS-14 dataset. We can observe that ASM-Loc generates more true positive predictions with high confidence scores and produces less localization error and confusion error (at the top $1G$ scoring predictions). It verifies that ASM-Loc improves the detection results by predicting more accurate action boundaries with our action-aware segment modeling modules.

Ablation on the increased receptive field. To further demonstrate that the effectiveness of our intra- and inter-segment attention modules is due to the segment-centric design instead of the increased receptive field, we replace our intra- and inter-segment attention modules with convolutional layers and compare the experimental results. From Table 8 we can see that by replacing the attention modules with convolutional layers, the performances drop by at least 3.3%, and even fall below the base model. We hypothesize that increasing the kernel size of the convolutional layers may lead to confusion between foreground and background snippets especially near the action boundaries. In contrast, our segment-centric attention design can model temporal structures within and across action segments and localize actions more precisely. The results verify that the segment-centric design is the key to our intra- and inter-segment attention modules.

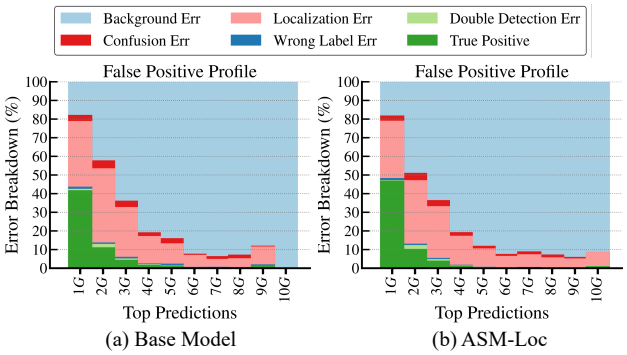


Figure 4. Diagnosing detection results. We present DETAD [1] false positive profiles of the base model and our ASM-Loc.

2. Action Proposal Generation

In Alg. 2, we present the details of how to generate action proposals $\tilde{\mathcal{S}}$ from the action localization results (*i.e.*, action segments) $\mathcal{S}$. Specifically, we first sort all the segment scores across the set $\mathcal{S}(c)$ for each ground-truth class c. Then we sum the confidence scores of all the action segments and output q_{sum}, and pick the top-K action segments with their confidence scores summation equal to $\alpha * q_{sum}$ to form the action proposals. Note that the number of the action proposals is video-adaptive and content dependent, despite α is shared for all videos. Finally, following the common practice in temporal action localization [2–5], we extend each proposal on both ends by δ of the proposal length to get an extended proposal with a longer temporal duration which can take more context-related snippets into consideration.

Algorithm 2: Action Proposal Generation

Input: Predicted Action Segments $\mathcal{S} = \{(s_i, e_i, c_i, q_i)\}_{i=1}^{I}$, selection ratio α, segment extension parameter δ
Output: Action Proposals $\tilde{\mathcal{S}} = \{(\tilde{s}_n, \tilde{e}_n, \tilde{c}_n)\}_{n=1}^{N}$

for *ground-truth class* c **do**
  $\mathcal{S}(c)_{sorted} \leftarrow \text{SORT}(\mathcal{S}(c))$ // sort segments by scores of class c
  $q_{sum} = \sum q_i$ // sum confidence scores for all segments
  Select K, s.t. $\max_K \sum_{i=1}^{K} q_i \leq \alpha * q_{sum}$ // select top-K segments from $\mathcal{S}(c)_{sorted}$
  $\tilde{\mathcal{S}}(c) : \{\tilde{s}_i, \tilde{e}_i, \tilde{c}_i\}_{i=1}^{K} = \{s_i - \delta(e_i - s_i), e_i + \delta(e_i - s_i), c_i\}_{i=1}^{K}$ // extend selected segments on both sides
end

Table 8. Ablation on the increased receptive field.

Modeling	Kernel Size	mAP@IoU (%)							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
Base	-	67.8	60.7	51.8	41.3	30.7	19.9	10.1	40.3
Conv	3	66.2	59.3	50.5	39.9	29.9	19.2	9.1	39.2
	5	66.5	58.9	51.0	40.0	29.7	19.3	9.8	39.3
	9	67.1	59.8	50.4	40.1	29.1	19.2	10.2	39.4
Attention	-	68.9	63.1	54.9	44.5	34	22.0	11.9	**42.7**

Table 9. Ablation on different action proposal selection methods.

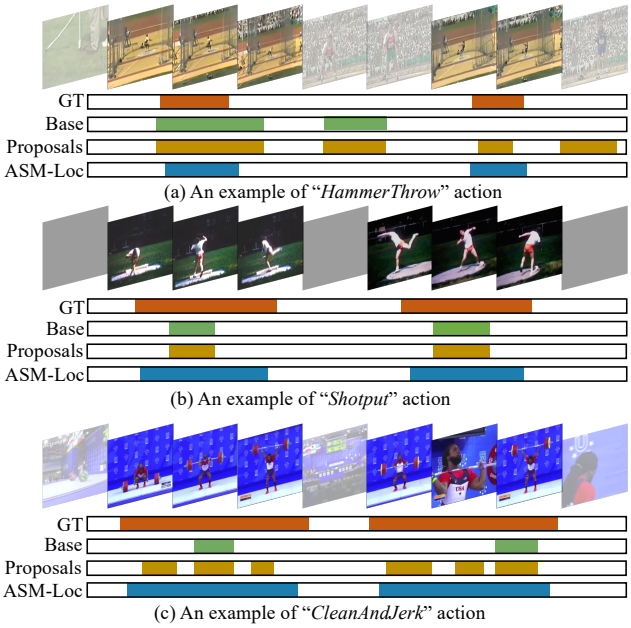
Method	mAP@IoU (%)							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
(a)	69.9	63.8	56	45.8	36.6	25.0	13.5	44.4
(b)	70.5	64.6	57.3	46.8	35.7	24.3	14.2	44.8
(c)	71.2	65.5	57.1	46.8	36.6	25.2	13.4	**45.1**

To verify the effectiveness of our proposal generation design, we compare three different settings of the segment selection procedure: **(a)** Fixed number of selected action segments where K is a fixed value for each class, which is not video-adaptive and content dependent; **(b)** K proportional to the number of predicted action segments in $\mathcal{S}(c)$, where $K = \alpha * |\mathcal{S}(c)|$; **(c)** our design. In Table 9, we can see that our design achieves the best results among the three designs.

3. Experiment Details

For the hyper-parameters, we set $\lambda_{\text{fg}} = 1, \lambda_{\text{bg}} = 0.5, \lambda_{\text{abg}} = 0.5, \beta = 0.2, \gamma = 6, H = 8, \delta = 0.5, \alpha = 0.7$ for THUMOS-14 and $\lambda_{\text{fg}} = 5, \lambda_{\text{bg}} = 0.5, \lambda_{\text{abg}} = 0.5, \beta = 0.2, \gamma = 10, H = 8, \delta = 0, \alpha = 0.3$ for ActivityNet-v1.3.

Following [6, 7], during inference, we use a set of thresholds to obtain the predicted action instances, then perform non-maximum suppression to remove overlapping segments. Specifically, for THUMOS-14, we set the foreground attention threshold from 0.1 to 0.9 with step 0.025, and perform NMS with a t-IOU threshold of 0.45. For ActivityNet-v1.3, we set the foreground-attention threshold from 0.005 to 0.02 with step 0.005, and apply NMS with a t-IoU threshold of 0.9.



(a) An example of "*HammerThrow*" action
(b) An example of "*Shotput*" action
(c) An example of "*CleanAndJerk*" action

Figure 5. Visualization of ground-truth, predictions and action proposals. Top-2 predictions with the highest confidence scores are selected for the base model and our ASM-Loc. Transparent frames represent background frames.

We implement our method in PyTorch [8] and train it on a single NVIDIA RTX1080Ti gpu.

4. More Qualitative Results

We provide more qualitative results in Figure 5. The first example of action "*HammerThrow*" shows the missed detection of short actions and over-completeness error. The second and third example of action "*Shotput*" and action "*CleanAndJerk*" shows the incompleteness error. It clearly shows that our ASM-Loc can help address these errors with more accurate action boundary predictions.

5. Limitation

The main limitation of our ASM-Loc is that the performance of our action-aware segment modeling modules depends on the generated action proposals. When the action proposals are largely misaligned with the ground-truth action segments, our ASM-Loc is not able to fix the error and generate correct predictions, as shown in Figure 3.

6. Broader Impacts

As the most popular media format nowadays, most information is spread in the format of videos. The temporal action localization task aims at finding the temporal boundaries and classifying category labels of actions of interest in untrimmed videos. Unlike supervised learning based approach that requires dense segment-level annotations, our proposed weakly-supervised temporal action localization model ASM-Loc only requires video-level labels. Therefore, WTAL is much more valuable in the real-world applications such as popular video-sharing social-network services, where billions of videos have only video-level user-generated tags. Besides, WTAL has broad applications in various fields, *e.g*. event detection, video summarization, highlight generation and video surveillance.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[2] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017. 1

[3] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[4] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. 1

[5] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11499–11506, 2020. 1

[6] Can Zhang, Meng Cao, Dongming Yang, Jie Chen, and Yuexian Zou. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16010–16019, 2021. 2

[7] Sanqing Qu, Guang Chen, Zhijun Li, Lijun Zhang, Fan Lu, and Alois Knoll. Acm-net: Action context modeling network for weakly-supervised temporal action localization. *arXiv preprint arXiv:2104.02967*, 2021. 2

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 2