Supplementary Material: Action Sensitivity Learning for Temporal Action Localization

Jiayi Shao[1], Xiaohan Wang[1], Ruijie Quan[1], Junjun Zheng[2], Jiang Yang[2], Yi Yang[1]
[1]ReLER Lab, CCAI, Zhejiang University, [2]Alibaba Group

This document provides more details of our approach and additional experimental results, as shown below:

- Details of Implementation and Architecture of ASL
- Additional Quantitative Results on ActivityNet1.3
- Additional Ablative Results
- Additional Qualitative Results

1. Details of Implementation and Architecture

The overall architecture is detailed in Table a. For each dataset, the training and architecture details are of little difference.

Respectively, for Multithumos [14], we use RGB-only I3D [3] pretrained on Kinetics to extract the video features. We upsample the input features to a fixed length of 1024 using linear interpolation and train the model with a batch size of 2, a learning rate of 0.0002, an epoch of 60 and a wight decay of 0.05.

For Charades [11], we use RGB-only I3D [3] model to extract the video features. We upsample the input features to a fixed length of 512 and train the model with a batch size of 32, a learning rate of 0.0004, an epoch of 15 and a weight decay of 0.05.

For Ego4D-Moment Queries v1.0 [7], we use EgoVLP [9], Slowfast [5] and Omnivore [6] network to extract the video features. We upsample the input features to a fixed length of 1024 and train the model with a batch size of 2, a learning rate of 0.0001, an epoch of 10 and a weight decay of 0.05. $l_1 = 2, l_2 = 3$. l_3 equals 8. The number of heads and embedding dimension d_{emb} are 8 and 512.

For Epic Kitchens 100 [4], we use Slowfast [5] features. We upsample the input features to a fixed length of 1024 and train the model with a batch size of 2, a learning rate of 0.0001, and a weight decay of 0.05 on noun and verb sub-task for 20 and 15 epochs respectively.

For Thumos14 [12], we use two-stream I3D [3] pretrained on Kinetics to extract the video features. We extend the input length to 1024 and train the model with a batch size of 2, a learning rate of 0.0001, an epoch of 30 and a weight decay of 0.05.

For ActivityNet1.3 [2], we use two-stream I3D [3] pretrained on Kinetics to extract the video features. We downsample the input features to a fixed length of 192 and train the model with a batch size of 16, a learning rate of 0.001, an epoch of 13 and a weight decay of 0.01. The number of heads and embedding dimension d_{emb} are 4 and 256.

For most datasets (if no additional noting), $l_1 = 1, l_2 = 2$. l_3 equals 5, l_4 equals 2, the number of heads and embedding dimension d_{emb} are 8 and 512. For all datasets, we use AdamW optimizer with a linear warmup and a cosine learning rate decay strategy. We present the pseudo-code of Action Sensitivity Learning (ASL) as shown in Algorithm 1.

Algorithm 1 The pseudo-code of ASL

Arguments:The labeled dataset $\mathcal{H} = \{V\}$, ground-truth instance $\mathcal{G} = \{\bar{t}^s, \bar{t}^e, \bar{c}\}$, Transformer Encoder $\mathcal{E}$, class-level action sensitivity p^{cls}, p^{loc} instance-level evaluator Φ^{cls}, Φ^{loc},localization head $\mathcal{D}_{loc}$, classification head $\mathcal{D}_{cls}$.

```
initialize h^cls, h^loc, E, Φ^cls, Φ^loc, D_loc, D_cls
for i ∈ [1, 2, ··· , N] do:
    Sample batch B ∈ H
    L ← 0
    for V in B do:
        f ← E(V)
        f_gt ← Sampling(f, (t̄^s, t̄^e))
        q^cls ← Φ^cls(f_g t)
        q^loc ← Φ^loc(f_g t)
        h^cls ← p^cls 𝟙[c̄] + q^cls
        h^loc ← p^loc 𝟙[c̄] + q^loc
        L ← L + h^loc L_loc
        L ← L + h^cls L_cls
        L ← L + L_s                          ▷ Defined in Eq.7.
    L ← L + L_ASCL                           ▷ Defined in Eq.14.
    Calculate ∂L
    Update h^cls, h^loc, E, Φ^cls, Φ^loc, D_loc, D_cls
return h^cls, h^loc, E, Φ^cls, Φ^loc, D_loc, D_cls
```

Table a. **The architecture of our model.** conv denotes 1-D convolution layers, where k is the kernel size, s is the stride, c_i, c_o is the input and outputfeatures dimensions. For Transformer-based parts, DS denotes Downsampling, self attn and channel attn is the normal self-attention operation on the temporal dimension and proposed channel attention operation on the channel dimension. GT from DS Transformer$_i$ denotes using ground-truth segments to sample features from outputs of DS Transformer$_i$. T_{GT} is the length of ground-truth segments. FC denotes fully connected layers.

	Name	Layer	Input	Output Size
	Input clip	-	-	T×D
encoder	Projection	conv $k=3, s=1(c_i=D, c_o=d_{emb})$	input clip	T $\times d_{emb}$
	TCN enc	$l_1\times$[conv $k=3, s=1(c_i=d_{emb}, c_o=d_{emb})$]	Projection	T $\times d_{emb}$
	Transformer enc	$l_2\times$ [[self attn + channel attn], [feedforward network]]	TCN enc	T $\times d_{emb}$
	DS Transformer$_i$, $(i=1,2,\cdots,l_3)$	$l_3\times$ [[self attn], [feedforward network]]	DS Transformer$_{i-1}$	$\frac{\text{T}}{2^{i-1}}\times d_{emb}$
Instance-level evaluator	Inst. evaluator	$l_4\times$ [conv $k=3, s=1(c_i=d_{emb}, c_o=d_{emb})$] FC $(c_i=512, c_o=1)$	GT from DS Transformer$_i$	$T_{GT}\times d_{emb}$ $T_{GT}\times 1$
heads	Cls or Loc heads	conv $k=3, s=1(c_i=d_{emb}, c_o=512)$ conv $k=3, s=1(c_i=512, c_o=512)$ conv $k=3, s=1(c_i=512, c_o=1$ or $2)$	DS Transformer$_i$	$\frac{\text{T}}{2^{i-1}}\times d_{emb}$ $\frac{\text{T}}{2^{i-1}}\times d_{emb}$ $\frac{\text{T}}{2^{i-1}}\times 1$ or $\frac{\text{T}}{2^{i-1}}\times 2$

Table b. **Additional Results on ActivityNet1.3**. We report mAP at different tIoU thresholds. Average mAP in [0.5:0.05:0.95] is reported on ActivityNet1.3. Best results are in **bold**.

Model	Feature	ActivityNet1.3 0.5	0.75	0.95	Avg.
AFSD [8]	I3D [3]	52.4	35.3	6.5	34.4
TadTR [10]	I3D [3]	49.1	32.6	8.5	32.3
Actionformer [15]	I3D [3]	54.2	36.9	7.6	36.0
Actionformer [15]	TSP [1, 13]	54.7	37.8	8.4	36.6
ASL	I3D [3]	54.1	37.4	8.0	36.2
ASL	TSP [1, 13]	**54.9**	**37.8**	**8.6**	**36.7**

Table c. **Additional Ablations on Thumos14.** *Class.* and *Inst.* means using class-level and instance-level action sensitivity learning.

method	avg mAP
baseline	66.08
baseline + Inst.	66.96
baseline + Class.	67.12
baseline + ASL	67.74
baseline + ASL + ASCL	67.88

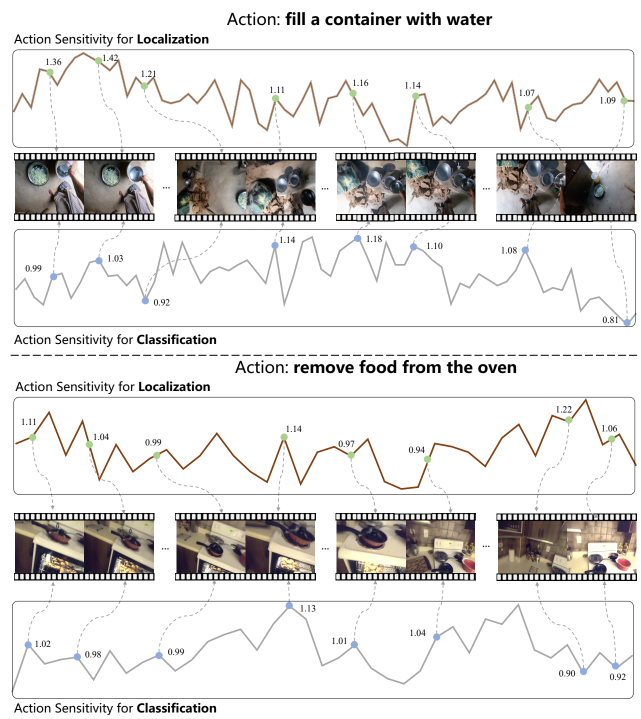

Figure a. Visualization of (**Top**) the frame sensitivity to sub-tasks of Action: **fill a container with water** and (**bottom**) Action: **remove food from the oven**. Please zoom in for the best view.

2. Additional Quantitative Results on ActivityNet1.3

[15] shows that using TSP features [1, 13] will benefit the performance on ActivityNet1.3 [2] more. We here report additional quantitative results on ActivityNet1.3 using TSP features. As shown in Table b, ASL also outperforms previous state-of-the-art methods using no matter I3D or TSP features, demonstrating the advantages of our approach.

3. Additional Ablative Results

This section provides additional ablative results on Thumos14 [12]. As shown in c, *baseline* denotes our base model without action sensitivity learning, our proposed action sensitivity learning and contrastive loss both boosts the performance of average mAP.

4. Addition Qualitative Results

In this section, we provide more qualitative results for action sensitivity learning. As shown in a, we provide qualitative results of action: *fill a container with water* and *re-*

move food from the oven. Frames involving main components of action (i.e. *water and pot*, *food*) are of a relatively high action sensitivity while those ambiguous and transitional frames are of a lower action sensitivity for both classification and localization sub-task. Meanwhile, sensitive frames may vary depending on the specific sub-tasks, in line with our decoupled design.

References

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2021. 2

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 1, 2

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1, 2

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 1

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[6] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *CVPR*, 2022. 1

[7] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[8] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3320–3329, June 2021. 2

[9] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *arXiv preprint arXiv:2206.01670*, 2022. 1

[10] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022. 2

[11] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 510–526. Springer, 2016. 1

[12] Yu-Gang Jiang&Jingen Liu&A Roshan Zamir&George Toderici&Ivan Laptev&Mubarak Shah& Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes. 2014. 1, 2

[13] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2017. 2

[14] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126:375–389, 2018. 1

[15] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, volume 13664 of *LNCS*, pages 492–510, 2022. 2