Pyramid Region-based Slot Attention Network for Temporal Action Proposal Generation (Supplementary Material)

Shuaicheng Li*[1]
lishuaicheng@sensetime.com
Feng Zhang*[2]
fzhang20@fudan.edu.cn
Rui-Wei Zhao[3]
rwzhao@fudan.edu.cn
Kunlin Yang[1]
yangkunlin@sensetime.com
Lingbo Liu[4]
liulingbo918@gmail.com
Rui Feng†[2]
fengrui@fudan.edu.cn
Jun Hou[1]
houjun@sensetime.com

[1] Sensetime Research
[2] Shanghai Key Laboratory of Intelligent Information Processing
Fudan University
Shanghai, China
[3] Academy for Engineering and Technology Fudan University
Shanghai, China
[4] The Hong Kong Polytechnic University
Hong Kong, China

Recall that the original slot attention is introduced to capture object-centric representations from raw perceptual input, the key idea of our PRSA-Net is to adopt Pyramid Region-based Slot Attention (PRSlot) to model action-matter representations. In this supplementary material, we include the following materials:

1) Further details on the proposed Pyramid Region-based Slot Attention (PRSlot).

2) Details about datasets and implementations.

3) Additional ablation studies.

4) More qualitative results.

1 More details of our proposed PRSA-Net

The original Slot Attention [15] is firstly proposed for object-centric representation learning, which is applied for unsupervised object discovery and image reconstruction. Slot attention

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

* Equal contribution
† Corresponding author

excels in abstracting centric representations from low-level features. With the iteration strategy, slot attention decomposes the input features into slot representations and specialize on the extraction of individual objects. Inspired by these, we attempt to take a natural next step to apply slot attention to video data and solve the temporal action proposal generation task, which aims to locate the start and end frame for an untrimmed video.

1.1 Insight of Region-based Attention mechanism.

Naturally, the action instance can be distinguished from the background due to the scene transitions. However, since untrimmed video contains complicated background and inaccurate action-related content in action instance, simply employing the similarity-based attention to update the action information may introduces spurious representations. To focus on the action-matter representation, we consider the boundary-aware features and action-matter features with the proposed region-based attention (RA) mechanism. RA summarize the local context for abstracting the slot representation. For boundaries, RA captures discrimination representation for the rapid motion from background to action starting. Also, for continuing action frame it extracts the progressive action transitions in local regions referred to as action-matter features.

1.2 Position Embedding

Since the output slot representations are permutation invariance in slot attention, it is essential to employ position embedding to order the video sequence. Different from the previous methods [5, 19], we introduce a position-based interaction matrix to capture position-matter contextual information. The inter-snippet connections are represented by an adjacency matrix A referred to as position attention. The input features calculate the weighted sum of the position attention and generate position-matter information which are referred to as *position embedding*.

1.3 Details of Baseline

As previous mentioned in the main paper, the baseline model with original attention achieve limited performance in proposal generation, we now describe the details of the baseline. When applying the off-the-shelf slot attention (please refer to the literature [15]), we found that it is hardly work. Therefore, based on it, we remove the MLP layers and replace the `WeightMean` operation with `WeightSum`. Additionally, we add the position embedding as described above. Even so, the baseline model only achieve limited performance listed in the main paper.

1.4 Action Deteciton on ActivityNet-1.3

Table 1 shows our action detection results on the validation set of ActivityNet-1.3 compared with previous works. Again, our method outperforms most other methods by a large margin in almost all cases, including the average mAPs over different tIoUs.

Method	0.5	0.75	0.95	Average
TAL-Net [1]	38.23	18.30	1.30	20.22
BSN [15]	46.45	29.96	8.02	30.03
P-GCN [23]	48.26	33.16	3.27	31.11
BMN [14]	50.07	34.78	8.29	33.85
G-TAD [22]	50.36	34.60	9.02	34.09
BSN++ [13]	51.27	35.70	8.33	34.88
BC-GNN [1]	50.56	35.35	9.71	34.68
RTD-Net [18]	47.21	30.68	8.61	30.83
Ours	**52.37**	**37.18**	**9.78**	**36.26**

Table 1: Comparison with state-of-the-arts detection methods on ActivityNet-1.3 in terms of mAP@tIoUs(%).

Position embedding	@50	@100	@200	@500
w/ o	47.5	55.1	60.3	61.8
w/ `learnable`	48.1	55.3	60.7	62.2
w/ `sine/cosine`	48.7	55.6	61.0	62.7
w/ `ours`	**49.1**	**56.1**	**61.3**	**63.2**

Table 2: Ablation study on position embedding. Evaluated on THUMOS14 in terms of AR@AN (%).

2 Details about datasets, metric and implementations

2.1 Datasets and Metric

THUMOS14 [7]. THUMOS14 dataset contains respectively 200 videos in the validation for training and 213 videos in the test set for inference. It has a total number of 20 classes, and each video has around 15 action instances on average.

ActivityNet-1.3 [2]. ActivityNet-1.3 dataset contains 19994 untrimmed videos labeled in a wider range of 200 action categories with a lower 1.5 action instances per video on average. These videos are split into the training, validation, and test set by the ratio of 2:1:1. We evaluate on the validation set of ActivityNet-1.3.

Evaluation metric. To assess the performances of action proposal generation, we report the average recall (AR) under different intersections over union thresholds (tIoUs) with various average number of proposals (AN) for each video. Following the conventions, we adopt the tIoUs of $\{0.5:0.05:1.0\}$ on THUMOS14 and $\{0.5:0.05:0.95\}$ on ActivityNet-1.3. To evaluate the quality of our generated proposals, we also evaluate the performances of action detection using the mean average precision (mAP) at different tIoUs. Following the official evaluation API, the tIoUs of $\{0.3,0.4,0.5,0.6,0.7\}$ are used for THUMOS14, and $\{0.5,0.75,0.95\}$ are used for ActivityNet-1.3.

2.2 Implementations

Following the conventional setting, we extracted the 2048-dimensional video features by two-stream I3D [3] pre-trained on Kinetics [9] on THUMOS14. Besides, for a fair comparison, we also conduct experiments based on the two-stream network TSN [20] backbone, where ResNet network [6] and BN-Inception network [8] are applied as the spatial and temporal network respectively. We also set $C_{\texttt{input}} = 256$. On ActivityNet-1.3, the video features were extracted by the pre-trained model provided in [2, 21]. In data preparation, we set the snippet interval σ to 4 on THUMOS14 and 16 on ActivityNet-1.3. Then we cropped each

Method	0.7	0.6	0.5	0.4	0.3
AFSD	**31.1**	43.7	55.5	62.4	67.3
AFSD w/ PRSlot	30.7	**44.5**	**57.4**	**63.5**	**69.3**

Table 3: Performance comparisons on THUMOS14 in terms of mAP@tIoUs (%). The A *w/* B denote A contains the module B.

video feature sequence with the overlapped windows of stride of 100 and length $L = 250$ on THUMOS14. While on ActivityNet-1.3, we set the temporal length to $L = 100$ by linear interpolation. Also, we enumerate all possible anchors where the max durations is $D = 64$ on THUMOS14 and $D = 100$ on ActivityNet-1.3. For the PRSlot module, we set the embed dimension $C_{\texttt{embed}} = 256$, $C_{\texttt{out}} = 256$, and the local region scale $S = \{4, 8\}$. In post-processing, we set the NMS threshold θ to 0.65 and 0.45 on THUMOS14 and ActivityNet-1.3 respectively. During the training, it was optimized on one NVIDIA TESLA V100 GPU with batchsize 16 on ActivityNet1.3 and 8 on THUMOS14. We adopted the Adam optimizer [10] for network optimization. For THUMOS14, the models were tuned for 10 epochs with the learning rate set to 2×10^{-4}. For ActivityNet-1.3, we trained our models by setting the learning rate to 10^{-3} in the first 7 epochs and decaying it to 10^{-4} in the last 3 epochs.

3 The generalization ability of PRSlot module

Different from directly using original video frames in our method, the TCA-Net [16] takes the original video and **proposals** generated by other two-stage methods, i.e., BSN [13] as inputs to refine the proposals and detect action instances. So TCA-Net [16] is out of the scope of our comparison. Even so, due to the concern about the generalization of our method, we conduct an additional experiment to evaluate our PRSlot module in the anchor-free method AFSD [11]. Specifically, we introduce our PRSlot module to augment the output of Feature extraction in AFSD [11]. Since the FPN is applied in Feature Extraction, we apply one PRSlot module for each level feature and output the augmented representation with the same sizes in AFSD. Finally, the augmented representation is processed to Coarse Prediction and Refined Prediction. As shown in Table 3, AFSD with an off-the-shelf PRSlot module achieves superior performance compared to the original AFSD [11]. This result verified the generalization of our PRSlot module. We will supplement the detailed implementation and experimental results in our supplementary material.

4 More Ablation Studies

We provide the additional ablation studies in this section and demonstrate the effectiveness of our model setting and compare our default setting to other reasonable alternatives. Similar to the main paper, We conduct experiments on THUMOUS14 using I3D [42] backbone and NMS algorithm.

4.1 Study on position embedding.

In Table 2, we show that employing different position embedding manner to input features at Pyramid Region-based Slot Attention module. It improves AR obviously using our position-

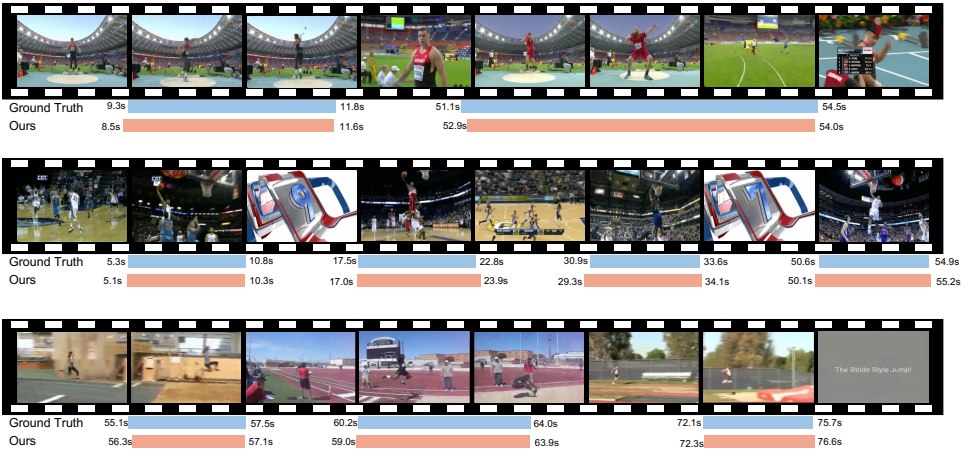

Figure 1: Some action proposal examples generated by our model on THUMOS14.

matter information, while applying the learnale position embedding [5] or *sine/cosine* absolute position embedding [19] could decrease the performance slightly. However, the performance decreases rapidly when we remove the position embedding, which demonstrate the effectiveness of the position embedding in video sequence.

5 More Qualitative Results

Here, we visualize more action detection results by our proposed PRSA-Net on THUMOS14. We can find that our model can successfully detect the boundaries of these action instances.

References

[1] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A Large-scale Video Benchmark for Human Activity Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[3] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[7] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS Challenge on Action Recognition for Videos "In the Wild". *Computer Vision and Image Understanding (CVIU)*, 155:1–23, 2017.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[9] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The Kinetics Human Action Video Dataset. *arXiv:1705.06950*, 2017.

[10] Diederik P Kingma and J Adam Ba. A method for stochastic optimization. *arXiv:1412.6980*, 2019.

[11] 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 *CVPR*, 2021.

[12] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *European Conference on Computer Vision (ECCV)*, 2018.

[13] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

[14] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[15] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33: 11525–11538, 2020.

[16] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, 2021.

[17] Haisheng Su, Weihao Gan, Wei Wu, Junjie Yan, and Yu Qiao. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. *arXiv preprint arXiv:2009.07641*, 2020.

[18] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13526–13535, 2021.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[20] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *European Conference on Computer Vision (ECCV)*, 2016.

[21] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. CUHK & ETHZ & SIAT Submission to ActivityNet Challenge 2016. *arXiv:1608.00797*, 2016.

[22] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph Convolutional Networks for Temporal Action Localization. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.