Supplementary Materials for UBoCo : Unsupervised Boundary Contrastive Learning for Generic Event Boundary Detection

Hyolim Kang [*], Jinwoo Kim [*], Taehyun Kim , Seon Joo Kim

Yonsei University

{hyolimkang,jinwoo-kim,kimth0101,seonjookim}@yonsei.ac.kr

1. Implementation Details

1.1. Feature Preprocessing

Initial features, which are taken into our custom encoder, are extracted from pretrained feature extractors. To be specific, we tested two feature extractors in our paper, including the image-level model, ImageNet [1] pretrained ResNet-50 [3] and the snippet-level model,Kinetics [4] pretrained TSN [9]. For competition[1] setting, we utilized additional feature extractors including kinetics pretrained SlowFast [2] network, yielding additional performance gain.

For all videos in the dataset, we preprocess them to be 24 fps so that each frame have the same temporal duration (1/24 seconds). After that, we sample the frames uniformly with the stride 6, making each feature have the temporal duration of 0.25 seconds. Since the play time of the video ranges from 0 to 10 seconds, we set the input features to have the length 40 so that the input features can represent maximum to 10 seconds. For the videos shorter than 10 seconds, we pad the features with the last feature vector. As a consequence, the input for our models have the shape of (40, 2048) with the ResNet-50 extractor, and (40, 4096) with the TSN extractor.

1.2. Model Architecture

For clear explanation, schematic diagram of our model's architecture is provided (Figure 1). Basically, we adopted multi-channel TSM approach, even though it is mean-pooled through channel dimension in UBoCo. Recall that Recursive TSM Parsing (RTP) algorithm, which is essential for UBoCo, cannot take multi-channel input.

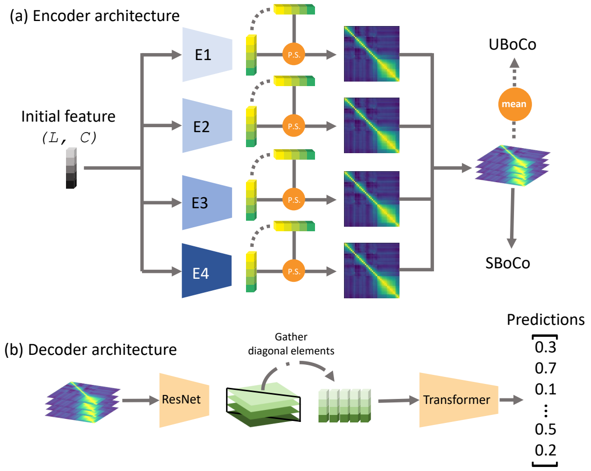

Figure 1. Detailed architecture of encoder/TSM decoder of UBoCo/SBoCo. Both UBoCo and SBoCo shares the same encoder architecture, but multi-channel TSM is average pooled for UBoCo (dotted line heading to "UBoCo"), whereas original multi-channel form is directly used for SBoCo. "E" sign in "E1, E2, E3, E4" stands for "Encoder", and different architecture is applied for each encoder.

For each encoder, we incorporated different architectures that have different "receptive field". For instance, we used 1d-convolutional neural network with kernel size 1 for short-term encoder E1, while Mixer [7] is applied for long-term encoder E4. For middle-term encoder E2 and E3, multi-layered 1d convolutional neural network with different depth is utilized, These various encoders with different receptive fields would catch different aspects of similarity among features, enriching the multi-channel TSM. Especially for long-term encoder, we conducted ablation study in Section 2.3.

For the TSM decoder, we used conventional ResNet [3]

[*] equal contribution, ordered by surname

[1] CVPR'21 LOng-form VidEo Understanding (LOVEU) Kinetics-GEBD challenge

Rel.Dis. threshold		0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	avg
Unsuper.	SceneDetect	27.5	30.0	31.2	31.9	32.4	32.7	33.0	33.2	33.4	33.5	31.88
	PA-Random	33.6	43.5	48.4	51.2	52.9	54.1	54.8	55.4	55.8	56.1	50.58
	PA	39.6	48.8	52.0	53.4	54.4	55.0	55.5	55.8	56.1	56.4	52.70
	(ours) UBoCo-Res50	70.3	83.9	86.2	88.5	88.9	89.3	89.4	89.8	90.0	90.2	86.65
	(ours) UBoCo-TSN	70.2	84.6	86.2	87.9	88.8	88.9	89.5	89.7	90.4	90.5	86.67
Super.	BMN	18.6	20.4	21.3	22.0	22.6	23.0	23.3	23.7	23.9	24.1	22.29
	BMN-StartEnd	49.1	58.9	62.7	64.8	66.0	66.8	67.4	67.8	68.1	68.3	63.99
	TCN-TAPOS	46.4	56.0	60.2	62.8	64.5	65.9	66.9	67.6	68.2	68.7	62.72
	TCN	58.8	65.7	67.9	69.1	69.8	70.3	70.6	70.8	71.0	71.2	68.52
	PC	62.5	75.8	80.4	82.9	84.4	85.3	85.9	86.4	86.7	87.0	81.73
	(ours) SBoCo-Res50	73.2	82.7	85.3	87.7	88.2	89.1	89.4	89.9	89.9	90.7	86.61
	(ours) SBoCo-TSN	78.7	86.0	88.4	90.5	90.7	90.7	91.1	91.7	91.8	92.2	89.18

Table 1. F1 results on Kinetics-GEBD for various unsupervised and superivsed GEBD methods including our UBoCo and SBoCo.

Gap	UBoCo-Res50	UBoCo-TSN
2	27.4	43.4
4	69.5	-
6	69.3	-
8	**70.3**	**70.2**
10	68.8	-
12	68.2	-
32	63.0	59.4

Table 2. F1@0.05 scores with various gap hyperparameter.

		Mean Difference	
		0.0	1.0
UBoCo-Res50	f1	68.1	70.3 (+2.2)
	precision	57.0	64.8 (+7.8)
	recall	84.5	76.8 (-7.7)
UBoCo-TSN	f1	68.1	70.2 (+2.1)
	precision	57.1	64.8 (+7.7)
	recall	84.4	76.6 (-7.8)

Table 3. Changes of the f1, precision, and recall scores according to the different *mean difference* values.

and Transformer [8] architecture. While other submodules are quite straightforward, note that only diagonal elements of the ResNet output, or feature map are taken as the input to transformer network (Figure 1 (b)). In the procedure, the tensor shape of the feature map turns into (B, C, L, L) to (B, C, L), where B stands for batch size, C means channel number, and L denotes the TSM's width/height.

2. Additional Experiments

In this section, we demonstrate additional ablation studies that justify our proposed approach. To reduce variance, all the experimental results, including results in the main paper, are the mean values of 5 different experiments with different random seeds.

Before we start, full table following [6]'s threshold convention is provided (Table 1) for clear demonstration of UBoCo / SBoCo's superior performance.

2.1. Local similarity prior

As we mentioned in the main paper, "gap" is the hyperparameter to materialize our local similarity assumption. Unlike semantic coherency prior, the local similarity assumption can be seen as nonessential since UBoCo can be conducted solely based on semantic coherency mask. To validate its functionality, we conducted an ablation study about the gap hyperparameter. The results can be seen in the Table 2 and as we expected, inappropriate gap size does not bring the best result. For an excessively large gap, there would be many improper positive pairs and trivial negative pairs, while a small gap results in too restrictive ones. Adequate gap size (in our experiment, 8) prevents this phenomena, yielding the best f1 score.

2.2. Mean Difference

Mean difference is the hyperparameter that forces the RTP algorithm to be terminated before the parsed TSM size is smaller than the predefined threshold T_1. (Detailed explanation can be found in the main paper's Section 3.2.1) The above termination condition is posed to make RTP not split the TSM that does not have any distinctive boundary score. With this, the RTP algorithm selects the boundaries more conservatively, suppressing false positives. Table 3 shows the experimental result of the ablation study about mean difference condition. As the algorithm being conservative, the precision score increases while the recall score

Long-term	BoCo loss	SBoCo-Res50	SBoCo-TSN
		71.7	77.6
✓		71.8 (+0.1)	77.5 (-0.1)
	✓	72.2 (+0.5)	78.1 (+0.5)
✓	✓	**73.2 (+1.5)**	**78.7 (+1.1)**

Table 4. F1 scores for the different combinations of long-term encoder layer and BoCo loss.

drops. However, as the balance between precision and recall gets better, we can achieve 2% improvement in the f1 score, the most important measure in our task.

2.3. Long-term Layer

To justify the utility of long-term encoder (E4 in Figure 1 (a)) and auxiliary BoCo loss in supervised setting, we conducted an ablation study about them. Overall result can be found in the Table 4. As the second row in the Table 4 shows, only adding the long-term layer does not improve the model performance, implying that the BCE loss alone is not sufficient to train the model with less inductive bias. On the other hand, sole BoCo loss brings meaningful improvement on the F1 score in both feature settings. (third row in Table 4) However, the best performance is achieved when both long-term encoder and BoCo loss are deployed (fourth row in Table 4), indicating that long-term encoder and auxiliary BoCo loss may have mutually beneficial relationship.

3. Additional Discussion

3.1. Claryfying BoCo Loss

For clarity, we explicitly put formal description of BoCo loss below. Let i_k and j_k denote a kth positive/negative sample's similarity score respectively, and m and n represent the number of positive/negative samples. Then, the contrastive loss term L_{contra} is defined as follows:

$$L_{contra} = \frac{1}{m}\sum_{k=1}^{m} j_k - \frac{1}{n}\sum_{k=1}^{n} i_k. \quad (1)$$

3.2. Cross Validation

To validate our method's generalization ability, we conducted slightly modified cross-validation on TAPOS dataset [5], the dataset for temporal action parsing. As event boundary annotations of TAPOS dataset does not coincide with GEBD concept, we cannot evaluate our method with TAPOS annotation. Instead, we **trained** our UBoCo model with *unlabeled TAPOS videos*, and **evaluated** the model with *Kinetics-GEBD validation videos and annotations*. Not surprisingly, we observed almost no performance drop (**69.2 F1@0.05**, ResNet50 feature), which supports our method's generalizability and transferability.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[2] 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*, 2019. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[4] 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 preprint arXiv:1705.06950*, 2017. 1

[5] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Intra-and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 730–739, 2020. 3

[6] Mike Zheng Shou, Stan W Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *ICCV*, 2021. 2

[7] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021. 1

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[9] 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 *ECCV*, 2016. 1