Supplementary Material: Temporal Cue Guided Video Highlight Detection with Low-Rank Audio-Visual Fusion

Qinghao Ye[1,2*†] Xiyue Shen[3*] Yuan Gao[4*] Zirui Wang[1*] Qi Bi[5] Ping Li[1†] Guang Yang[6]
[1] Hangzhou Dianzi University [2] University of California, San Diego [3] East China Normal University
[4] University of Oxford [5] Wuhan University [6] Imperial College London
{16205234,hdu_wzr,lpcs}@hdu.edu.cn; q7ye@ucsd.edu; 51205901080@stu.ecnu.edu.cn;
yuan.gao2@eng.ox.ac.uk; 2009biqi@163.com; g.yang@imperial.ac.uk

1. Implementation Details

We implement the proposed method in PyTorch platform with SGD optimizer. We use a batch size of 1024. The initial learning rate is set 0.01 and scaled by a factor of 0.1 for every 50 epochs. The L_2 weight decay coefficient is set to 5×10^{-4}. We empirically set $T = 2$ for both YouTube and TVSum dataset, and the training procedure is terminated after 200 epochs. All experiments were conducted on a machine with a single NVIDIA TITAN RTX GPU.

For a fair comparison, we reproduced and reported the results of the MINI-Net [5], which also utilizes the auditory and visual information, with their officially released codes and trained on our self-collected dataset. Following the protocol widely used in [6, 5], we trained the model on self-collected dataset, then evaluate on the benchmark datasets (*i.e*., YouTube Highlights and TVSum). In particular, Xiong *et al*. [6] collected approximate 10 million videos from Instagram for training, and Hong *et al*. [5] trained MINI-Net with their self-collected approximate 200k videos with 8k videos per topic through contacting with the authors since they did not mention these in the paper. However, those datasets are not publicly available in which we doubt with their actual performance. For this reason, following the same protocol [6, 5], we crawled about 35k videos (average of 1.4k videos per topic) based on hashtags from Instagram as training set .

Besides, we sample the topic-specific videos that are shorter than 60 seconds as positive videos, and take the video longer than 60 seconds in videos with different tags as negative videos. To preprocess each video, we break a video up uniformly into one-second clips and randomly sample the consecutive clips ($\tau = 60$ clips) during training.

For audio and visual feature extraction, we use 3DResNet-34 network [3] pretrained on Kinetics-600 dataset [1] to extract the visual feature $\mathbf{f}_v \in \mathbb{R}^{512}$, and the audio feature $\mathbf{f}_a \in \mathbb{R}^{128}$ is extracted by VGGish model [4] pretrained on AudioSet dataset [2].

Besides, we follow the standard evaluation protocol as [6, 5], *i.e*., the mean average precision is utilized to measure the model performance on YouTube Highlight dataset, and Top-5 mean average precision for TVSum dataset.

2. Computation Issue

In this part, we investigate the computation efficiency for audio-visual tensor fusion module. Assume the inputted video segment embedded features are denoted as $\mathbf{f}_v \in \mathbb{R}^{d_v}$ and $\mathbf{f}_a \in \mathbb{R}^{d_a}$. Then the time complexity of computing fused features $\mathbf{f}_h \in \mathbb{R}^{d_h}$ with core tensor $\mathcal{T}_c$ is $O(d_v d_a d_h)$ since $\mathbf{f}_h = (\mathcal{T}_c \times_1 \mathbf{f}_v) \times_2 \mathbf{f}_a$. For simplicity, we set $d_v = d_a = d_h = d$, so the time complexity of above equation is $O(d^3)$.

However, the low-rank audio-visual tensor fusion module of our approach only requires $O(Rd^2)$, which is faster than the original version since $R \ll d$. Therefore, our method is more efficient compared with the method without low-rank constraint.

3. More Quantitative Results

We vary the rank constraints R to the audio-visual feature fusion. As explained earlier, we use a series of rank one kernels to decompose the multi-modal feature representation $\mathcal{T}_c$ that we argues by doing his, we project the fused features into an unimodal subspace. We found it is crucial to choose the right amount of constraints to balance the complexity of decomposition while maintaining the useful interactions between video and audio features. As we can see in Figure 1, there is a considerable gain in the detection performance as stronger constraint putting into place until the critical point *i.e*., $R = 8$ that excessive constraints will no longer help and the performance starts to drop moderately.

To further validate the benefit of combining multiple modalities, we trained the models without audio features or visual features, and the testing results are summarized in Table 1. From the table, we can observe that even only trained with visual features, our method is still able to achieve com-

*Equal contribution.
†Corresponding authors.

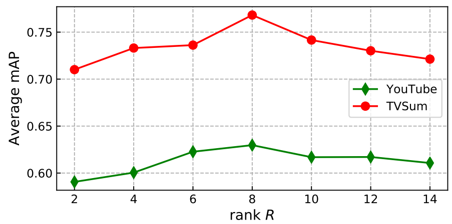


Figure 1. Variations in performance by constrain the rank R of core tensor $\mathcal{T}_c$.

Methods	YouTube	TVSum
MINI-Net* w/o vision	0.4853	0.5474
MINI-Net* w/o audio	0.5539	0.6675
MINI-Net* [5]	0.5837	0.7020
Ours w/o vision	0.5316	0.6041
Ours w/o audio	0.5892	0.7364
Ours	**0.6297**	**0.7682**

Table 1. Performance comparison of different models with single modality and dual-modalities on two datasets. * indicates our implementation trained on self-collected dataset.

Number of Clips	YouTube	TVSum
$\tau = 20$	0.6173	0.7427
$\tau = 40$	0.6208	0.7415
$\tau = 60$	**0.6297**	**0.7682**
$\tau = 80$	0.6223	0.7440

Table 2. Performance comparison of different number of clips in training process on two datasets.

parable results than MINI-Net [5]. Besides, it can be observed that visual features contribute to the model's performance more significantly than the audio features.

We also investigate the effect of sampled video lengths for training. In Table 2, it can be observed that the performance varies not significantly with different number of sampled clips during the training procedure, and we get the best performance when $\tau = 60$.

In addition, we investigate the influence of L_1 regularization coefficient (β) to the variance margin $\mathcal{X}_{var}$ and the score margin $\mathcal{X}_s$, and the results are shown in Table 3. It shows that the smaller sparsity regularization coefficient tends to produce better performance.

Moreover, we conduct experiment on the CoSum dataset in addition to the YouTube Highlights and TVSum datasets. The CoSum dataset contains 51 videos with 10 different topics. The experimental results are summarized in Table 4. We can observe that our method achieves 0.9304 mAP, which outperformed other state-of-the-art methods. However, since CoSum dataset is dominated by single scenes that can be easily detected, the improvement would not be significant.

β	YouTube	TVSum
0.0001	**0.6297**	**0.7682**
0.001	0.6272	0.7603
0.01	0.6033	0.7195
0.1	0.6035	0.7161

Table 3. Average mAP comparison under different scale of sparsity regularization β on two datasets.

4. Supplementary Proof

In this section, we provide mathematical proof of our video score fusion scheme in attention-gated instance aggregation module in order to alleviate gradient vanishing problem encountered during positive video optimization.

We revisit the conventional Noise-OR video score aggregation method as:

$$\hat{p}_{\mathcal{V}}^{(i)} = 1 - \prod_{s=1}^{m}(1 - p_s^{(i)}), \tag{1}$$

where $p_s^{(i)}$ is the confidence score for segment $v_s^{(i)}$ of video $\mathcal{V}^{(i)}$. Then, we consider the optimization for positive videos, and take the partial derivative of binary cross entropy loss $\mathcal{L}$ as follows:

$$\frac{\partial \mathcal{L}}{\partial p_j^{(i)}} = \frac{\partial \mathcal{L}}{\partial \hat{p}_{\mathcal{V}}^{(i)}} \left(\prod_{k=1,k\neq j}^{m} (1 - p_k^{(i)}) \right) = \frac{\hat{p}_{\mathcal{V}}^{(i)} - 1}{\hat{p}_{\mathcal{V}}^{(i)} (1 - p_j^{(i)})}. \tag{2}$$

Then, provided that we pick two arbitrary segments $u, v \in [1, m]$ in the positive video, and set $p_u^{(i)} = \epsilon, p_v^{(i)} = 1 - \epsilon$, where $\epsilon \to 0$. In addition, the rest $p_j^{(i)}$ are set to $\delta \in (0, 1)$. The video score in Eq.(1) is computed as:

$$\hat{p}_{\mathcal{V}}^{(i)} = 1 - \prod_{j=1}^{m}(1 - p_j^{(i)}) = 1 - \epsilon(1 - \epsilon)\delta^{m-2}, \tag{3}$$

which indicates that $\hat{p}_{\mathcal{V}}^{(i)} \to 1$. Therefore, we can conclude that $\partial\mathcal{L}/\partial p_j^{(i)} \to 0$ from Eq.(2), which results in the gradient vanishing issue during the optimization.

By contrast, in the attention-gated instance aggregation module, we define video score as:

$$p_s = \sigma(W_p \mathbf{c}_s^{(T)} + b_p), \tag{4}$$

$$\hat{p}_{\mathcal{V}}^{(i)} = \sigma \left(W_p \sum_{j=1}^{m} \alpha_j \mathbf{c}_j^{(T)} + b_p \right), \tag{5}$$

where $\sigma(\cdot)$ is the normalized function represented as $\sigma(x) = 1/(1 + \exp(-x))$. For simplicity, we set $\alpha_i =$

Topic	Supervised Methods					Weakly Supervised Methods							
	KVS	DPP	sLSTM	SM	SMRS	Quasi	MBF	CVS	SG	DSN	VESD	MINI-Net*	Ours
Base Jump	0.662	0.672	0.683	0.692	0.504	0.561	0.631	0.658	0.698	0.715	0.685	0.7992	**0.8292**
Bike Polo	0.674	0.682	0.701	0.722	0.492	0.625	0.592	0.675	0.713	0.746	0.714	**0.9421**	0.9228
Eiffel Tower	0.731	0.744	0.749	0.789	0.556	0.575	0.618	0.722	0.759	0.813	0.783	0.9087	**0.9339**
Excavators River Cross	0.685	0.694	0.717	0.728	0.525	0.563	0.575	0.693	0.729	0.756	0.721	0.9866	**1.0000**
Kids Play in Leaves	0.701	0.705	0.714	0.745	0.521	0.557	0.594	0.707	0.729	0.772	0.742	0.9838	**1.0000**
MLB	0.668	0.677	0.714	0.693	0.543	0.563	0.624	0.679	0.721	0.727	0.687	0.9645	**0.9756**
NFL	0.671	0.681	0.681	0.727	0.558	0.587	0.603	0.674	0.693	0.737	0.724	1.0000	**1.0000**
Notre Dame Cathedral	0.698	0.704	0.722	0.759	0.496	0.617	0.594	0.702	0.738	0.782	0.751	0.9636	**0.9772**
Statue of Liberty	0.713	0.722	0.721	0.766	0.525	0.551	0.624	0.715	0.743	0.794	0.763	0.8833	**0.9058**
Surf	0.642	0.648	0.653	0.683	0.533	0.562	0.603	0.647	0.681	0.709	0.674	0.7394	**0.7598**
Average	0.684	0.692	0.705	0.735	0.525	0.576	0.602	0.687	0.720	0.755	0.721	0.9171	**0.9304**

Table 4. Performance comparison (Top-5 mAP score) on CoSum dataset. Our method outperforms against all of the compared state-of-the-art methods. * indicates our implementation trained on self-collected dataset.

$1 (i = 1, \cdots, m)$ and $b_p = 0$. Therefore, we can reform the video score as follows:

$$\begin{aligned} \hat{p}_{\mathcal{V}}^{(i)} &= \frac{1}{1 + \exp(-W_p \sum_{j=1}^{m} \mathbf{c}_j^{(T)})} \\ &= \frac{1}{1 + \prod_{j=1}^{m} exp(-W_p \mathbf{c}_j^{(T)})} \\ &= \frac{1}{1 + \prod_{j=1}^{m} (\frac{1}{\sigma(W_p \mathbf{c}_j^{(T)})} - 1)} \\ &= \frac{1}{1 + \prod_{j=1}^{m} (\frac{1}{p_j^{(i)}} - 1)}. \end{aligned} \tag{6}$$

Similarly, the gradient of binary cross entropy loss $\mathcal{L}$ can be computed as:

$$\frac{\partial \mathcal{L}}{\partial p_j^{(i)}} = \frac{\hat{p}_{\mathcal{V}}^{(i)} - 1}{p_j^{(i)} (1 - p_j^{(i)})}. \tag{7}$$

Without loss of generality, taking the same settings mentioned above, we can observe:

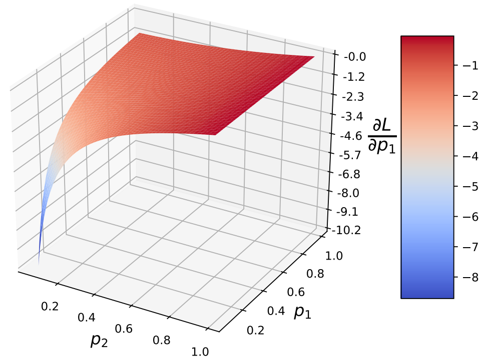
$$\hat{p}_{\mathcal{V}}^{(i)} = \frac{1}{1 + \prod_{j=1}^{m} (\frac{1}{p_j^{(i)}} - 1)} = \frac{1}{1 + (\frac{1}{\delta} - 1)^{m-2}}, \tag{8}$$

where $\delta \in (0, 1)$ is a constant, which shows that $\hat{p}_{\mathcal{V}}^{(i)} \nrightarrow 1$. As a consequence, $\partial \mathcal{L} / \partial p_j^{(i)} = \frac{1}{1 + (\frac{1}{\delta} - 1)^{m-2} (\delta(1-\delta))} \nrightarrow 0$.

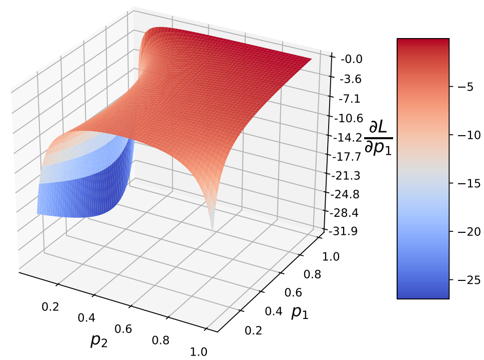
Besides, we also represent the visualization of binary case between original Noise-OR method and our proposed method in Figure 2. The area that gradient vanishes for Noise-OR method is much larger than that of our proposed method, which verifies the fact that our method can ease the gradient vanishing problem.

5. Visual Examples

In addition, we also illustrate the highlight detection results in Figure 3.



(a) Visualization of the gradient of Noise-OR method in binary case.



(b) Visualization of the gradient of our improved approach in binary case.

Figure 2. The visualization of gradient for two different instance aggregation schemes. (Best viewed in color.)

References

[1] 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] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal,

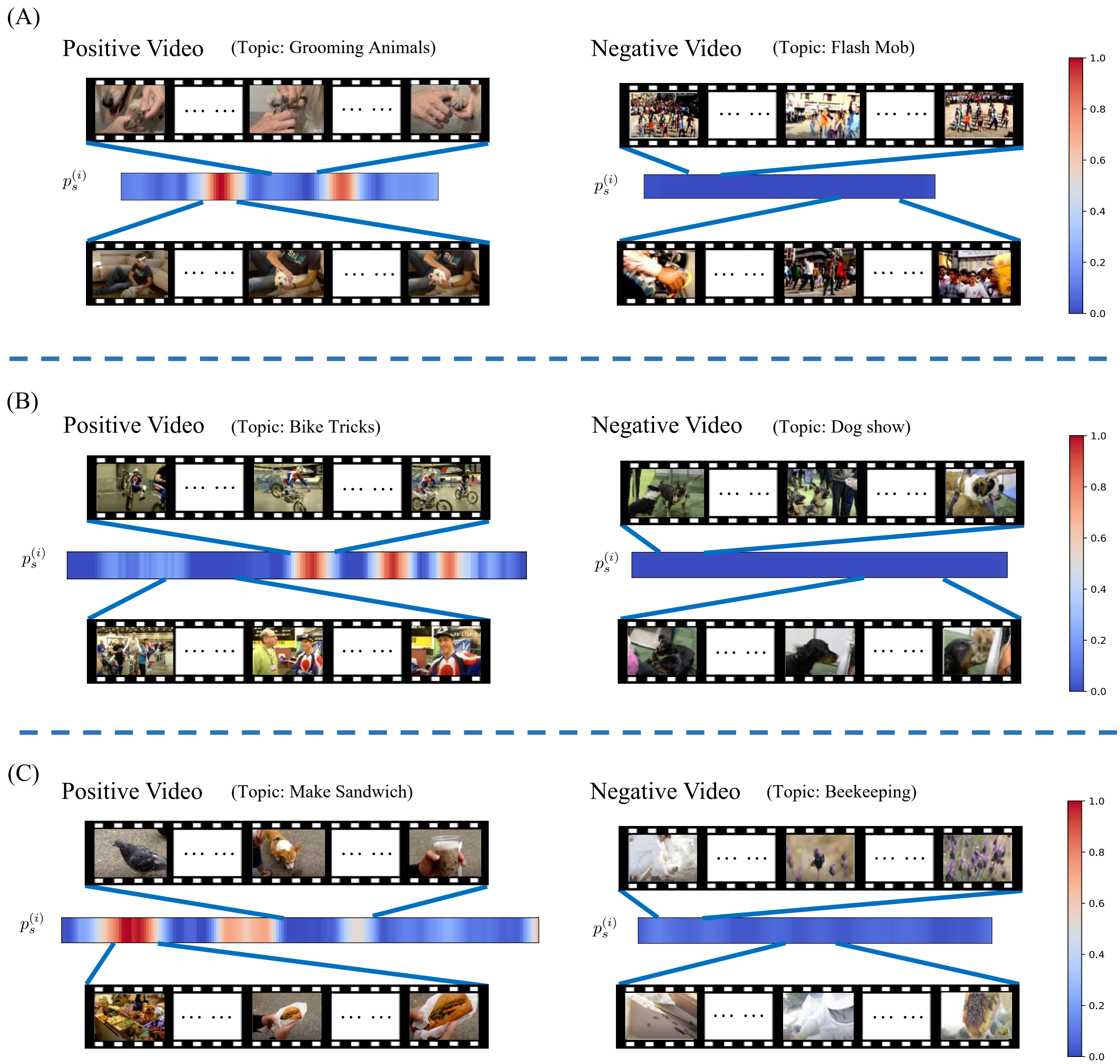


Figure 3. Qualitative results of our method on highlight detection. (Best viewed in color.)

and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780. IEEE, 2017. 1

[3] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 1

[4] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017. 1

[5] Fa-Ting Hong, Xuanteng Huang, Wei-Hong Li, and Wei-Shi Zheng. Mini-net: Multiple instance ranking network for video highlight detec- tion. In *European Conference on Computer Vision*, 2020. 1, 2

[6] Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1258–1267, 2019. 1