Supplementary Material
Divide and Conquer for Single-frame Temporal Action Localization

Chen Ju[1], Peisen Zhao[1], Siheng Chen[1], Ya Zhang[1✉], Yanfeng Wang[1], Qi Tian[2]
[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University [2]Huawei Cloud & AI
{ju_chen, pszhao, sihengc, ya_zhang, wangyanfeng}@sjtu.edu.cn, tian.qi1@huawei.com

1. Detailed Network Architectures

In our framework, the *seedframe detector*, the *location estimator*, the *gate-approximation network* in the *mask generator*, and the *classifier* are four key components. Table 1 details their network architectures. For simplicity, we denote a temporal convolutional layer as $conv(h_k, h_f)$, where h_k and h_f are the kernel size and the filter number of the convolutional layer. Similarly, a fully connected layer is denoted as $fc(f_a, f_b)$, where f_a and f_b are the input dimension and the output dimension. A dropout layer is denoted as drop(α), where α is the dropout rate.

For the *seedframe detector*, the input is the feature $\mathbf{X} \in \mathbb{R}^{T\times D}$ of the whole video, and the output is the estimated seedframe heatmap $\widehat{\mathbf{k}} \in \mathbb{R}^{T}$, where T and D are the video length and the feature dimension, respectively.

For the *location estimator*, the input is the feature $\mathbf{X}_{\rm s} \in \mathbb{R}^{T_{\rm s}\times D}$ of the video clip, and the output is the estimated proposal $\mathbf{v} = (\Delta p + p, \ell) \in \mathbb{R}^2$, where $T_{\rm s}$, p, Δp, and ℓ are the length of video clips, the timestamp of the seedframe, the center offset, and the action length, respectively.

For the *gate-approximation network* in the *mask generator*, the input is the estimated proposal $\mathbf{v} \in \mathbb{R}^2$, and the output is an approximate gate-shaped mask $\widehat{\mathbf{m}} \in [0, 1]^{T_{\rm s}}$.

For the *classifier*, the input is the clip-level foreground feature $\mathbf{x}_{\rm fg} \in \mathbb{R}^D$ (the clip-level background feature $\mathbf{x}_{\rm bg} \in \mathbb{R}^D$), and the output is the clip-level foreground classification probability $\widehat{\mathbf{y}}_{\rm fg} \in \mathbb{R}^{C+1}$ (the clip-level background-aware probability $\widehat{\mathbf{y}}_{\rm bg} \in \mathbb{R}^{C+1}$), where C denotes the total number of action categories and the additional one denotes the background category.

Table 1. Detailed network architectures. $conv(h_k, h_f)$ is a temporal convolutional layer, where h_k and h_f are the kernel size and the filter number, respectively. $fc(f_a, f_b)$ is a fully connected layer, where f_a and f_b denote the input dimension and the output dimension. drop(α) is a dropout layer with the dropout rate of α. And 'product' denotes the product operation.

Input	Layer	Activation	Output
Seedframe Detector			
$\mathbf{X}$ $(T \times D)$	$conv(3, 1024)$	ReLU	$\mathbf{X}_1$ $(T \times 1024)$
$\mathbf{X}_1$	$conv(3, 512)$	ReLU	$\mathbf{X}_2$ $(T \times 512)$
$\mathbf{X}_2$	$conv(3, 256)$	ReLU	$\mathbf{X}_3$ $(T \times 256)$
$\mathbf{X}_3$	$conv(3, 64)$	ReLU	$\mathbf{X}_4$ $(T \times 64)$
$\mathbf{X}_4$	$conv(3, 1)$	Sigmoid	$\widehat{\mathbf{k}}$ (T)
Location Estimator			
$\mathbf{X}_{\rm s}$ $(T_{\rm s} \times D)$	$fc(T_{\rm s}, T_{\rm s})$	ReLU	$\mathbf{X}_{\rm s1}$ $(T_{\rm s} \times D)$
$\mathbf{X}_{\rm s1}$	drop(0.5)	-	$\mathbf{X}_{\rm s2}$ $(T_{\rm s} \times D)$
$\mathbf{X}_{\rm s2}$	$fc(T_{\rm s}, 1)$	ReLU	$\mathbf{X}_{\rm s3}$ $(1 \times D)$
$\mathbf{X}_{\rm s3}$	$conv(3, 512)$	ReLU	$\mathbf{X}_{\rm s4}$ (1×512)
$\mathbf{X}_{\rm s4}$	$conv(3, 64)$	ReLU	$\mathbf{X}_{\rm s5}$ (1×64)
$\mathbf{X}_{\rm s5}$	$conv(3, 8)$	ReLU	$\mathbf{X}_{\rm s6}$ (1×8)
$\mathbf{X}_{\rm s6}$	$conv(3, 1)$	Sigmoid	$\mathbf{X}_{\rm s7}$ (1×1)
$\mathbf{X}_{\rm s6}$	$conv(3, 1)$	Tanh	$\mathbf{X}_{\rm s8}$ (1×1)
$\mathbf{X}_{\rm s7}$ $T_{\rm s}$	product	-	ℓ (1)
$\mathbf{X}_{\rm s8}$ $T_{\rm s}/16$	product	-	Δp (1)
Gate-Approximation Network			
$\mathbf{v}$ (2)	$fc(2, 32)$	ReLU	$\mathbf{v}_1$ (32)
$\mathbf{v}_1$	$fc(32, 64)$	ReLU	$\mathbf{v}_2$ (64)
$\mathbf{v}_2$	$fc(64, T_{\rm s})$	ReLU	$\mathbf{v}_3$ $(T_{\rm s})$
$\mathbf{v}_3$	$fc(T_{\rm s}, T_{\rm s})$	ReLU	$\mathbf{v}_4$ $(T_{\rm s})$
$\mathbf{v}_4$	$fc(T_{\rm s}, T_{\rm s})$	Sigmoid	$\widehat{\mathbf{m}}$ $(T_{\rm s})$
Classifier			
$\mathbf{x}_{\rm fg}$ (D)	$fc(D, D)$	ReLU	$\mathbf{x}_{\rm fg1}$ (D)
$\mathbf{x}_{\rm fg1}$	drop(0.7)	-	$\mathbf{x}_{\rm fg2}$ (D)
$\mathbf{x}_{\rm fg2}$	$fc(D, C+1)$	Sigmoid	$\widehat{\mathbf{y}}_{\rm fg}$ $(C+1)$

2. More Implementation Details

Our method is implemented with PyTorch [6]. Following previous literature [3, 4, 5], we split each untrimmed video into 16-frame snippets (In this paper, we use a frame to indicate such a snippet for simplicity), and use the pre-trained feature extractor without fine-tuning for fair comparison. To deal with the large variation in the video length, we sample T consecutive frames from each video during training. And T is set to 2500, 360, and 128 for THUMOS14, BEOID, and GTEA datasets. During testing, we feed all the video frames to the proposed framework. For all datasets, we utilize Adam optimizer with a learning rate of 10^{-4} to train the seedframe detector. To reduce the high-frequency noise in the estimated seedframe heatmap, we use the Savitzky-Golay filter for smoothing. In the video clip generation, we

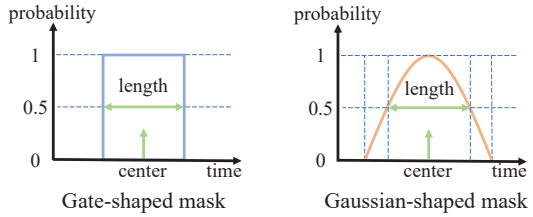

Figure 1. Detailed implementation of replacing the Gate-shaped mask with a Gaussian-shaped mask.

rescale the length of video clips to T_s frames. T_s is set to 128, 64, and 32 on THUMOS14, BEOID, and GTEA. To train the location estimator and the classifier, we use Adam with a learning rate of 10^{-4} for all datasets.

In the mask generator, we propose two solutions to handle the non-differentiable issue. One is to replace the Gate-shaped mask with a Gaussian-shaped mask. Figure 1 illustrates its detailed implementation. We set the action center of the proposal as the center of the Gaussian-shaped mask, and set the action length of the proposal as the full width at half maximum (FWHM) of the Gaussian-shaped mask.

The other is to approximate Eq. (3) in the main paper using a gate-approximation network. We first simulate 0.1 million proposals, each indicating the action center and the action length. The proposal simulation is achieved by uniformly sampling two-dimensional vector sets. And the action center and the action length are both fixed in the range of 0 to T_s. Then, for each proposal, we calculate the corresponding gate-shaped temporal mask as the ground-truth label. Finally, using Adam optimizer with a learning rate of 10^{-5}, we optimize the gate-approximation network to ensure that it accurately transforms the proposal into the T_s-dimensional approximate gate-shaped mask.

3. More Experimental Results

In this section, we carry out more ablation experiments on THUMOS14 dataset for further analysis.

Impact of the threshold θ. In the video clip generation, we mine the seedframes of action instances through the filtering threshold θ and local maxima. A larger threshold will generate fewer seedframes, thus omitting some action instances. While a smaller threshold will produce more seedframes, thus over-segmenting action instances. Figure 2 illustrates the impact. Our method achieves the best performance using the threshold 0.15. The performance fluctuations are small when θ is in the range of 0.05 to 0.25.

Impact of the hyperparameter β. The hyperparameter β is used to balance the foreground classification loss and the background-aware loss. Figure 3 demonstrates the results using different values of β. When β is in the range of 1 to 1.75, our method shows good robustness. However, when β is smaller than 1, the excessive background-aware loss misleads the model to identify most video regions as background, causing a rapid drop in performance. Similarly, when β is larger than 2, the excessive foreground classification loss causes many video regions to be identified as actions, also resulting in the performance degradation.

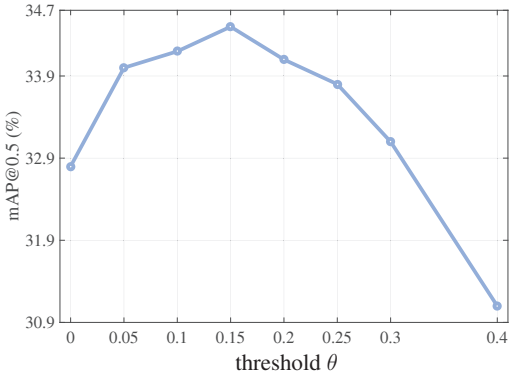

Figure 2. Impact of the filtering threshold θ.

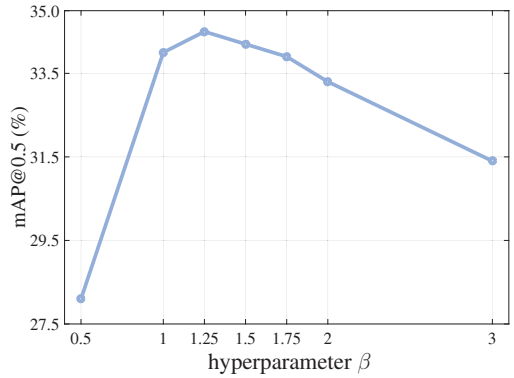

Figure 3. Impact of the trade-off hyperparameter β.

Table 2. Experiments on whether to freeze the weights of the gate-approximation network in the location estimation stage. AVG denotes the average mAP at IoU thresholds 0.1:0.1:0.7.

Freeze Weights	mAP@IoU 0.3	mAP@IoU 0.5	mAP@IoU 0.7	AVG
no	56.0	31.3	10.2	42.1
yes	**58.1**	**34.5**	**11.9**	**44.3**

Whether to freeze the gate-approximation network. In the mask generator, we adopt a learnable network to approximate the gate-shaped mask. The gate-approximation network is first trained independently of the location estimation stage, then its weights are frozen to ensure the accurate and robust transformation during the end-to-end training of the location estimation stage. Table 2 reveals the necessity of freezing the weights. If we do not freeze the weights of the gate-approximation network, but jointly optimize the network in the location estimation stage, the performance

Table 3. Comparison of simulated single-frame supervision. AVG denotes the average mAP at IoU thresholds 0.1:0.1:0.7.

Method	Distribution	mAP@IoU			AVG
		0.3	0.5	0.7	
SF-Net [3]	Manual	53.3	28.8	9.7	40.6
	Uniform	52.0	30.2	11.8	40.5
	Gaussian	47.4	26.2	9.1	36.7
Ours	Manual	58.1	34.5	11.9	44.3
	Uniform	55.6	32.3	12.3	42.9
	Gaussian	58.2	35.9	12.8	44.8

Table 4. Comparison with the state-of-the-art methods on ActivityNet1.2. Our method surpasses the competitor. AVG denotes the average mAP at IoU thresholds 0.5:0.05:0.95.

Supervision	Method	mAP@IoU			AVG
		0.5	0.7	0.9	
Full	CDC [8]	45.3	-	-	23.8
	SSN [11]	41.3	30.4	13.2	28.3
Weak Video-level	UNet [10]	7.4	3.9	1.2	3.6
	AutoLoc [9]	27.3	17.5	6.8	16.0
	WTALC [7]	37.0	14.6	-	18.0
	CMCS [2]	36.8	-	-	22.4
	3C-Net [4]	37.2	23.7	9.2	21.7
Weak Single-frame	SF-Net [3]	37.8	24.6	10.3	22.8
	Ours	**40.5**	**26.4**	**10.7**	**24.5**

will drop 2.2% average mAP. In this case, the network gradually loses the temporal smoothness constraints, and assigns unequal aggregation weights to action-related frames, resulting in the performance degradation.

Comparison of simulated single-frame labels. In addition to manually annotated single-frame labels, SF-Net [3] also provides two types of simulated single-frame labels, which are sampled from the ground-truth boundary labels via a uniform distribution and a Gaussian distribution. Table 3 compares our method with SF-Net using three types of single-frame labels. No matter what type of single-frame labels is used, our method surpasses the competitor, revealing the effectiveness and robustness. Notably, the single-frame labels generated by the Gaussian distribution are quite unrealistic and expensive, since they require annotators to have some knowledge of the action boundaries.

4. Comparison on ActivityNet Dataset

To verify the effectiveness of single-frame supervision on more diverse datasets, SF-Net [3] also randomly simulates single-frame annotations on ActivityNet1.2 [1]. This dataset contains 9682 videos belonging to 100 action categories, which are divided into 4819 videos for training, 2383 videos for validating, and 2480 videos for testing. It is a large-scale dataset, and each video contains an average of 1.5 action instances. The conventional choice is to train on the training set and evaluate on the validation set. Table 4 summarizes the comparison results. Following standard protocols, we use the mAP at different thresholds (0.5:0.05:0.95) for evaluation. It can be observed that our method outperforms the competitor [3] at all IoU thresholds, and follows the fully-supervised methods with the least gap, revealing the superiority of our method.

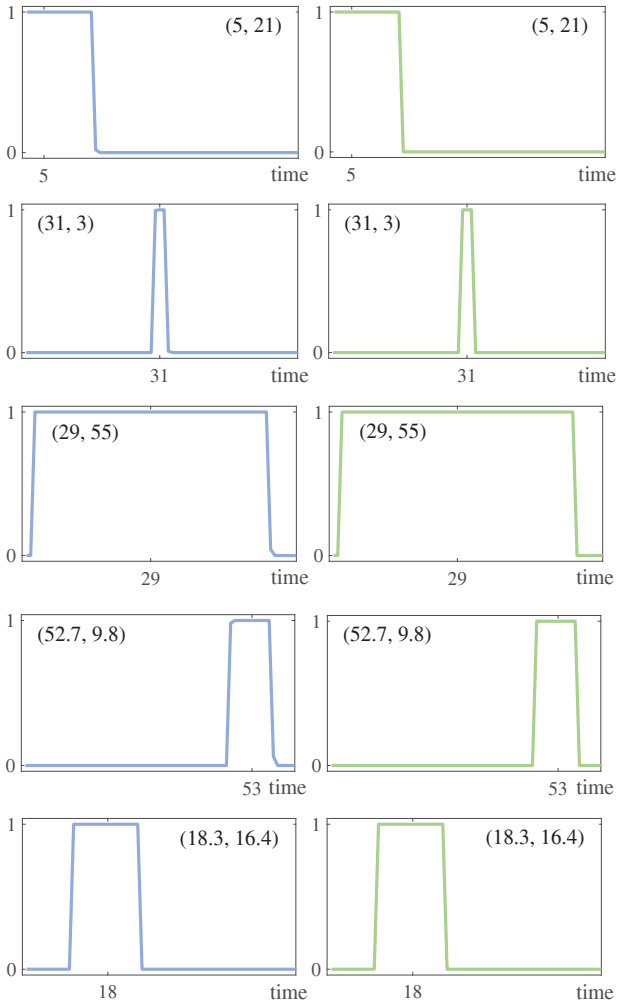

Figure 4. Qualitative results of the gate-approximation network. The left column denotes the output masks of the network, and the right column denotes the ground-truth masks. '(ξ, ζ)' denotes an action proposal, where ξ and ζ are the action center and the action length. For various types of proposals, the output masks of the network are almost the same as ground-truth gate-shaped masks, indicating the effectiveness of the gate-approximation network.

5. Qualitative Results

Figure 4 presents some qualitative results to intuitively demonstrate the superiority of the gate-approximation network in the mask generator. In these five cases, there are extremely long actions and extremely short actions. For var-

ious types of input proposals, the output masks of the gate-approximation network are almost the same as ground-truth gate-shaped masks, revealing the robustness and effectiveness of the gate-approximation network.

6. Future Work

This paper proposes a novel two-stage framework with the spirit of divide and conquer for single-frame temporal action localization. And there are several improvements left for future work. For example, dealing with two key challenges in real-world scenarios: actions with significantly varying timescales and overlapping actions.

To alleviate varying timescales, in the instance counting stage, we rescale (normalize) the length of video clips so that the extreme short/long clip is expanded/compressed. But the proportion of action instances may still vary in different video clips. In the location estimation stage, feature pyramids can be explored to make it easier to estimate the action length. On the other hand, overlapping actions cause multiple seedframe peaks to approach each other, making it unstable and difficult to pick local maxima. One feasible solution is to sample multiple sets of seedframes as multiple hypotheses, then generate action proposals respectively, and finally filter or sort these proposals.

References

[1] 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 (CVPR)*, pages 961–970, 2015.

[2] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1298–1307, 2019.

[3] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou. Sf-net: Single-frame supervision for temporal action localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 420–437, 2020.

[4] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8679–8687, 2019.

[5] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5502–5511, 2019.

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in neural information processing systems (NIPS)*, 2017.

[7] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[8] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5734–5743, 2017.

[9] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018.

[10] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4334, 2017.

[11] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2914–2923, 2017.