Supplemental Material - Robust Action Segmentation from Timestamp Supervision

Yaser Souri*[1, 2]
yasersouri@microsoft.com
Yazan Abu Farha*[1, 3]
yabufarha@birzeit.edu
Emad Bahrami*[1]
bahrami@iai.uni-bonn.de
Gianpiero Francesca[4]
gianpiero.francesca@toyota-europe.com
Juergen Gall[1]
gall@iai.uni-bonn.de

[1] Computer Vision Group
University of Bonn
Bonn, Germany

[2] Microsoft
Redmond, United States

[3] Birzeit University
Birzeit, West Bank, Palestine

[4] Toyota Motor Europe
Brussels, Belgium

* indicates equal contribution

We provide further details of the optimization, additional ablation studies, and report the runtime.

1 Optimization

As discussed in the paper, we optimize the objective:

$$\begin{aligned}\sum_{i=1}^{N}\left(\sum_{t=1}^{T} -log\tilde{y}_t[y_{p_i}]\mathcal{I}(t|p_i-l_i \leq t \leq p_i+r_i)\right)&\\ +\beta\sum_{t=1}^{T}\left(1-\sum_{i=1}^{N}\mathcal{I}(t|p_i-l_i \leq t \leq p_i+r_i)\right)&\end{aligned} \tag{1}$$

As the indicator function $\mathcal{I}$ is a non-differentiable function, we replace it with the differentiable plateau function from [2, 3]. The plateau function shown in Figure 1 is defined by

$$f(t|\lambda^c,\lambda^w,\lambda^s) = \frac{1}{(e^{\lambda^s(t-\lambda^c-\lambda^w)}+1)(e^{\lambda^s(-t+\lambda^c-\lambda^w)}+1)}. \tag{2}$$

It defines a window of size $2\lambda^w$ at the center λ^c. The parameter λ^s of the plateau function controls the sharpness of the transition from 0 to 1.

For optimization, we replace the indicator function $\mathcal{I}$ by the plateau function f:

$$\mathcal{I}(t|p_i-l_i \leq t \leq p_i+r_i) = f(t|\lambda^{c_i},\lambda^{w_i},\lambda^s) \tag{3}$$

where $\lambda^{c_i} = p_i + \frac{r_i-l_i}{2}$, $\lambda^{w_i} = \frac{r_i+l_i}{2}$, and $\lambda^s = 0.025$ is fixed. Equation (1) is thus re-written as

$$\sum_{i=1}^{N}\left(\sum_{t=1}^{T} -log\tilde{y}_t[y_{p_i}]f(t|\lambda^{c_i},\lambda^{w_i},\lambda^s)\right)+\beta\sum_{t=1}^{T}\left(1-\sum_{i=1}^{N}f(t|\lambda^{c_i},\lambda^{w_i},\lambda^s)\right). \tag{4}$$

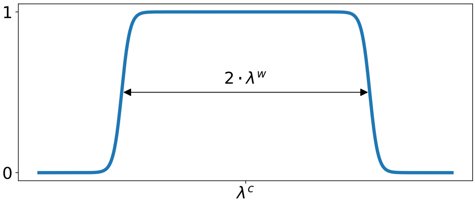

Figure 1: The plateau function (2) with center parameter λ^c and width parameter λ^w.

% Segments	Method	F1@{10, 25, 50}			Edit	Acc
95%	Uniform-2	63.1	56.4	37.8	58.8	59.5
	Uniform-3	63.4	58.5	40.8	56.9	63.5
	Timestamps only	59.9	55.2	45.6	49.6	71.5
	Ours	**72.9**	**69.6**	**57.5**	**64.2**	**75.3**
90%	Uniform-2	60.8	53.0	34.7	56.0	56.1
	Uniform-3	62.0	56.3	39.2	56.1	61.5
	Timestamps only	55.4	51.4	40.2	46.0	69.6
	Ours	**70.0**	**65.1**	**55.2**	**62.1**	**75.4**
80%	Uniform-2	56.2	49.3	32.1	51.1	56.3
	Uniform-3	59.6	52.5	35.3	54.4	59.7
	Timestamps only	55.1	50.8	39.6	44.8	66.2
	Ours	**70.9**	**67.8**	**53.7**	**61.4**	**73.1**
70%	Uniform-2	42.2	36.0	19.0	40.1	45.8
	Uniform-3	48.8	43.2	28.5	46.0	54.1
	Timestamps only	46.6	41.4	30.2	39.3	60.0
	Ours	**64.1**	**59.2**	**44.8**	**56.9**	**70.8**

Table 1: Comparison with different baselines on the 50Salads dataset.

For the gradient descent based optimization of (4), we initialize r_i, g_i, l_{i+1} uniformly, *i.e.*, $r_i = g_i = l_{i+1}$ and $r_i + g_i + l_{i+1} = p_{i+1} - p_i$. We optimize (4) for 30 iterations using the Adam optimizer with a learning rate of 0.03.

2 Additional Ablation Studies

2.1 Comparison with Baselines

We compare our optimization approach with a few baselines. The first baseline uses only the annotated timestamps for training and ignores all the frames in between, which is denoted by “Timestamps only". The second baseline “Uniform-2" divides the frames between the timestamps equally into two segments and assigns labels to each frame based on the label of the nearest timestamp. Whereas in the last baseline “Uniform-3", the frames between timestamps are divided into three equally sized segments. In this baseline, only the first and last segments are labeled by the corresponding timestamp and the middle segment is ignored during training. Results for our approach and the baselines on the 50Salads dataset are shown

% Segments	Initialization	F1@{10, 25, 50}			Edit	Acc
95%	Fixed (3 sec)	69.7	66.9	55.3	62.4	73.3
	Uniform	**72.9**	**69.6**	**57.5**	**64.2**	**75.3**
90%	Fixed (3 sec)	68.4	**65.7**	**55.3**	58.5	72.9
	Uniform	**70.0**	65.1	55.2	**62.1**	**75.4**
80%	Fixed (3 sec)	66.2	63.1	50.7	57.6	71.1
	Uniform	**70.9**	**67.8**	**53.7**	**61.4**	**73.1**
70%	Fixed (3 sec)	62.0	58.5	44.3	53.5	67.0
	Uniform	**64.1**	**59.2**	**44.8**	**56.9**	**70.8**

Table 2: Impact of initialization on the 50Salads dataset.

in Table 1. Our approach outperforms all baselines.

2.2 Impact of Initialization

As discussed in Section 1, we initialize r_i, g_i, l_{i+1} uniformly (Uniform-3). To analyse the impact of the initialization of the optimization, we compare it to another initialization where we set l_i and r_i to 3 seconds and $g_i = p_{i+1} - p_i - r_i - l_{i+1}$. Table 2 shows the results of the uniform initialization compared to the initialization based on a fixed duration. The uniform initialization scheme performs better.

2.3 Evaluation of Different Timestamps Selection Strategies

The timestamps provided by [□] follow a uniform distribution. We also analyze the performance if the timestamps follow a Gaussian distribution. To this end, we randomly sampled a timestamp for each ground-truth action from a Gaussian distribution using the center of the action as the mean and half of the duration of the action as the standard deviation. If the sample is outside the action, we set it to the start or end frame of the action, respectively. We also consider the case where the timestamps are at the center of each action and the worst case where all timestamps are at the beginning of each action. As pointed out in the supplemental

Method	Timestamps	F1@{10, 25, 50}			Edit	Acc
Li *et al.* [□]	Start frame	49.7	36.8	14.8	49.8	41.5
	Center frame	69.5	65.6	48.5	61.8	66.6
	Gaussian	67.1	62.5	45.4	58.0	66.3
	Uniform	63.9	59.6	44.3	57.6	63.8
Ours	Start frame	54.1	40.7	17.3	52.8	44.8
	Center frame	71.5	68.9	56.9	63.2	72.4
	Gaussian	70.8	67.2	55.4	62.3	71.9
	Uniform	70.0	65.1	55.2	62.1	75.4

Table 3: Results for different setups for providing timestamps. We use 90% of the timestamps on the 50Salads dataset.

Method	50Salads				
	F1@{10, 25, 50}			Edit	Acc
Li *et al.* [28]	64.7	60.1	47.1	57.1	67.5
Ours	**65.3**	**61.1**	**49.8**	**58.3**	**71.0**
Oracle	74.2	72.4	62.6	64.8	75.8

Table 4: Results if segments that are difficult to recognize by the network are missed. The results are reported on split 1 of the 50Salads dataset for 95% of the timestamps.

material of [□], humans would not annotate the start frame since it is more ambiguous. Table 3 shows that our approach outperforms [□] regardless of how the annotated timestamps are provided.

Finally, we evaluate a setup where action segments that are difficult to recognize by the network are more likely to be missed. To identify these segments, we trained a model using all timestamps for 30 epochs and used it to compute the average probability of the correct class for each ground-truth action segment. We set the sampling probability of a timestamp proportional to the inverse of the class probability of the corresponding ground-truth segment, i.e., timestamps with a low prediction probability are less likely to be sampled. We then sampled 95% of the action segments without replacement. We report the results in Table 4.

2.4 Unknown Frames

In the paper, we have already analyzed the impact of β on the accuracy. Figure 2 shows the average value of g_i (average length of an ignore region) and how often $g_i = 0$ (length zero) for different values of β. The results are reported for the training set of split 1 of the 50Salads dataset. As expected, the average size of g_i decreases as the value of β increases. Furthermore, we see that, even for large values of β, it occurs rarely that $g_i = 0$. This is desirable since there is usually a transition between two actions that should not be labeled by any of the two actions.

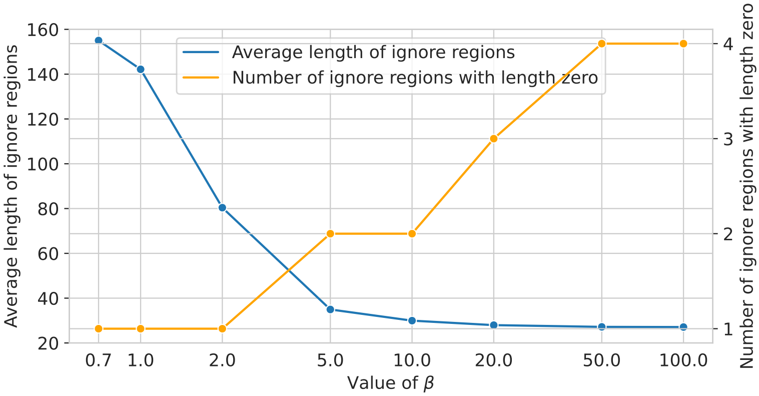

Figure 2: Average length of the ignore regions (average value of g_i) and number of the ignore regions with length 0 ($g_i = 0$) for different values of β. The numbers are reported for the training set of split 1 of the 50Salads dataset.

3 Runtime Comparison

Our proposed approach for generating labels from timestamps is not only more robust than [1], but it is also much faster. We measured the wall clock time for the whole training set of split 1 of the 50Salads dataset. While [1] requires 116 seconds to generate the labels, our approach requires only 1.7 seconds, which is 68 times faster.

References

[1] Zhe Li, Yazan Abu Farha, and Juergen Gall. Temporal action segmentation from timestamp supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8365–8374, 2021.

[2] Davide Moltisanti, Sanja Fidler, and Dima Damen. Action recognition from single timestamp supervision in untrimmed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9915–9924, 2019.

[3] Yaser Souri, Yazan Abu Farha, Fabien Despinoy, Gianpiero Francesca, and Juergen Gall. FIFA: Fast Inference Approximation for Action Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2021.