
ECCV'20
ONLINE
23-28 AUGUST 2020
16TH EUROPEAN CONFERENCE ON
COMPUTER VISION
WWW.ECCV2020.EU

Tencent AI Lab

Boundary-Aware Cascade Networks for Temporal Action Segmentation

Zhenzhi Wang[1], Ziteng Gao[1], Limin Wang[1],Zhifeng Li[2], and Gangshan Wu[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China

[2] Tencent AI Lab, Shenzhen, China

Temporal action segmentation

- Densely predict action labels for each video frame in long untrimmed videos
  - Action detection: output action labels for sparse proposals
- Action categories are fine-grained, e.g., actions in kitchen
  - Action detection: diverse actions with great difference

Motivation

Two existing challenges:

- Low accuracy near action boundaries
- Over-segmentation errors inside action instances

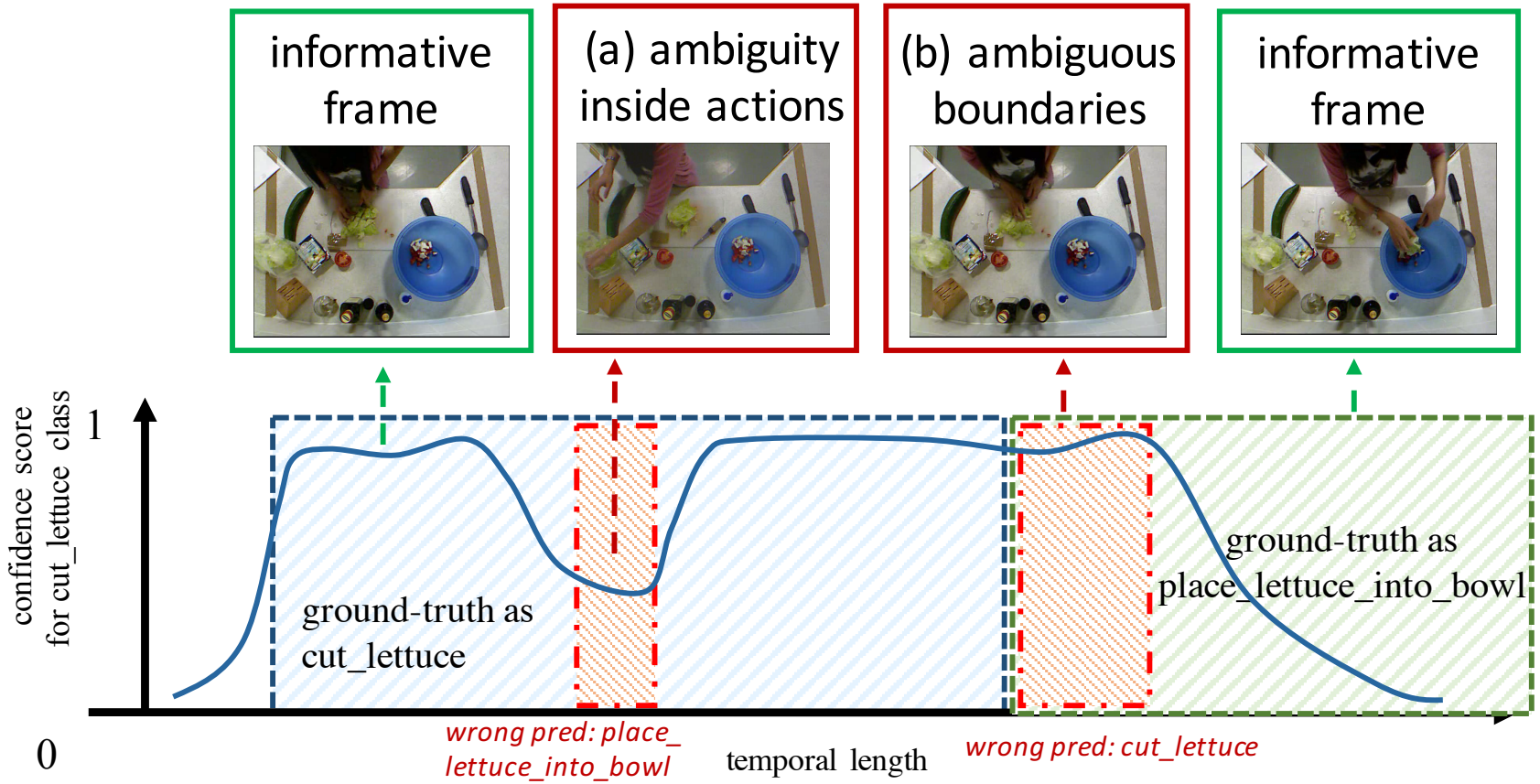


Fig.1: Illustration of two challenges

Previous works

Previous focus: capturing long-term dependency via larger receptive field using temporal convolution:

- Encoder-decoder structure: ED-TCN [1], TDRN [2]
- Dilated convolution: Dilated TCN [1], MS-TCN [3]

However, all of them use fixed modeling capacity, thus in meanwhile

- overfit simple frames -> over-segmentation errors
- struggle with hard frames, e.g., ambiguities frames near action boundaries

[1] Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR (2017)
[2] Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: CVPR (2018)
[3] Farha, Y.A., Gall, J.: MS-TCN: multi-stage temporal convolutional network for action segmentation. In: CVPR (2019)

Method: Stage Cascade

- Low accuracy near action boundaries

  -> propose a cascade paradigm to boost segmentation accuracy

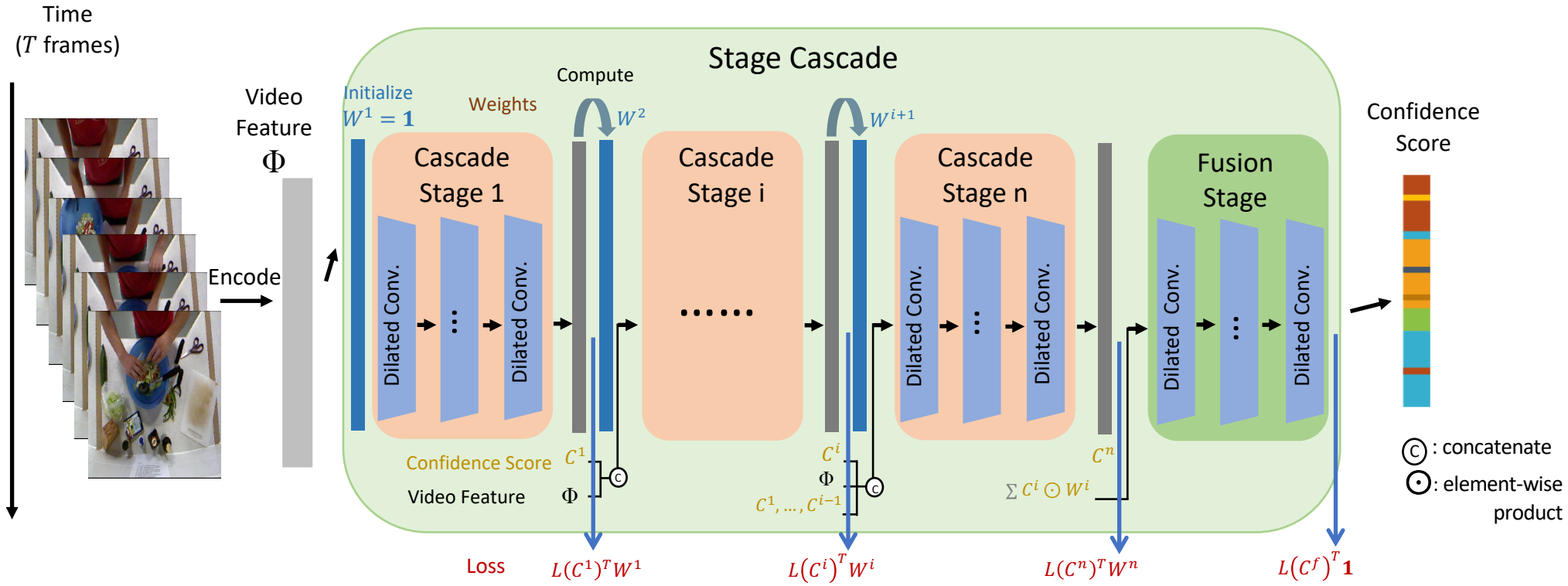


Fig.2: Stage Cascade

Method: Stage Cascade

Compute weights of i^{th} cascade stage and t^{th} frame

$$w_t^i = \begin{cases} e^{-c_t^{i-1}} w_t^{i-1} & \text{if } \; c_t^{i-1} \geq \rho, \\ e^{c_t^{i-1}} w_t^{i-1} & \text{if } \; \forall j \leq i-1, c_t^j < \rho \\ w_t^{i-1} & \text{if } \; c_t^{i-1} < \rho \; \text{ and } \; \exists j < i-1, c_t^j \geq \rho \end{cases}$$

Adapt the loss of each cascade stage from the weights to enable dynamic capacity

$$\mathcal{L}_{baseline} = \frac{1}{T} \sum_t -\log\left(y_{t,c}\right), \quad \mathcal{L}_i^{SC} = -\frac{\sum_t w_t^i \cdot \log\left(y_{t,c}\right)}{\sum_t w_t^i}$$

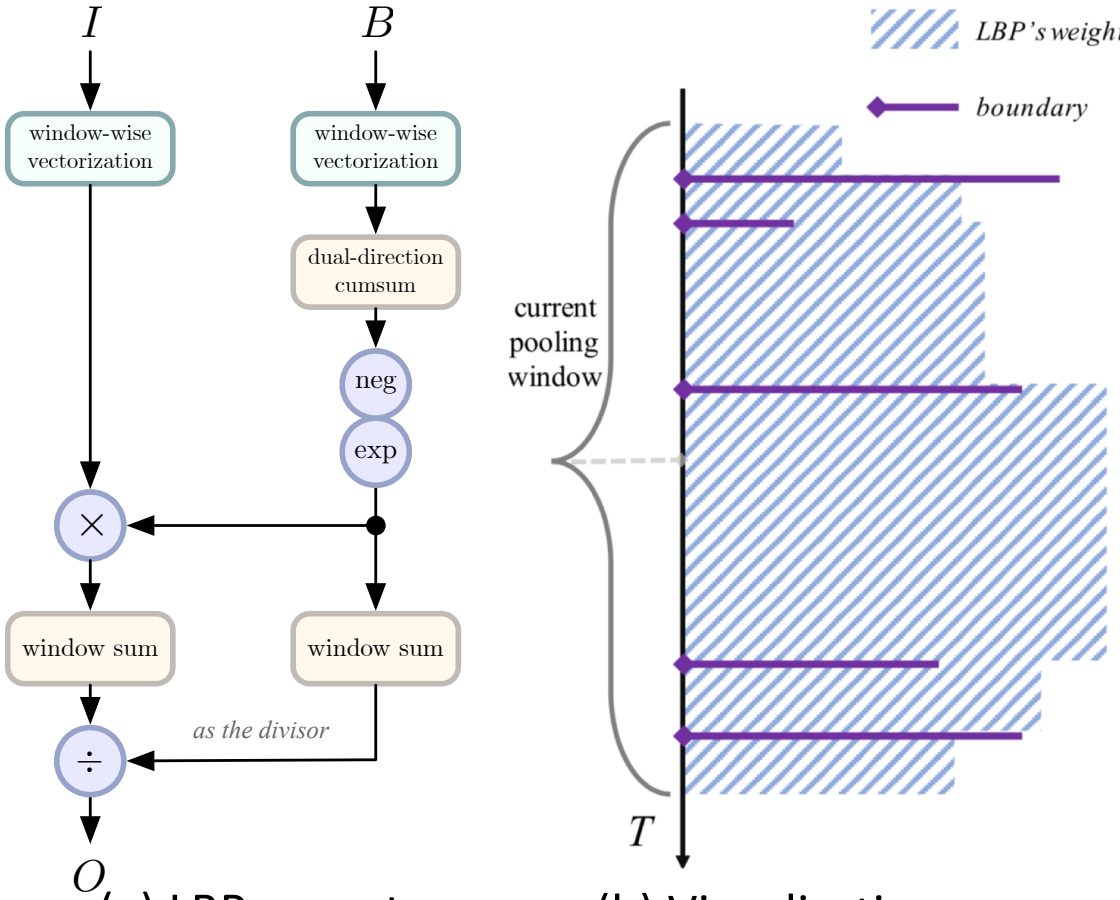
Aggregate the output of cascade stages from the weights, to be the input of fusion stage

$$c_t^f = \frac{\sum_i w_t^i c_t^i}{\sum_i w_t^i},$$

Method: Local Barrier Pooling

- Over-segmentation errors inside action instances

  -> propose a novel smoothing operator using semantic boundary information



(a) LBP operator (b) Visualization

Fig.3: Local Barrier Pooling

$$y'_{t,c} = \frac{y_{t,c} + \sum_{s\in\{-1,+1\}} \sum_{\beta=1}^{L} y_{t+s\cdot\beta,c} \exp(-\alpha \sum_{j=1}^{\beta} b_{t+s\cdot j})}{1 + \sum_{s\in\{-1,+1\}} \sum_{\beta=1}^{L} \exp(-\alpha \sum_{j=1}^{\beta} b_{t+s\cdot j})}$$

Two heuristic smoothing method are LBP's special cases

- Average Pooling (all barriers as 0)
- Gaussian-like smoothing (all barriers as 1)

Overall Framework

Training procedure:

- Pretrain BGM using boundary GT constructed from original GT
- Jointly train BGM and SC only using segmentation GT, BGM can also be optimized by backward gradients due to differentiable LBP

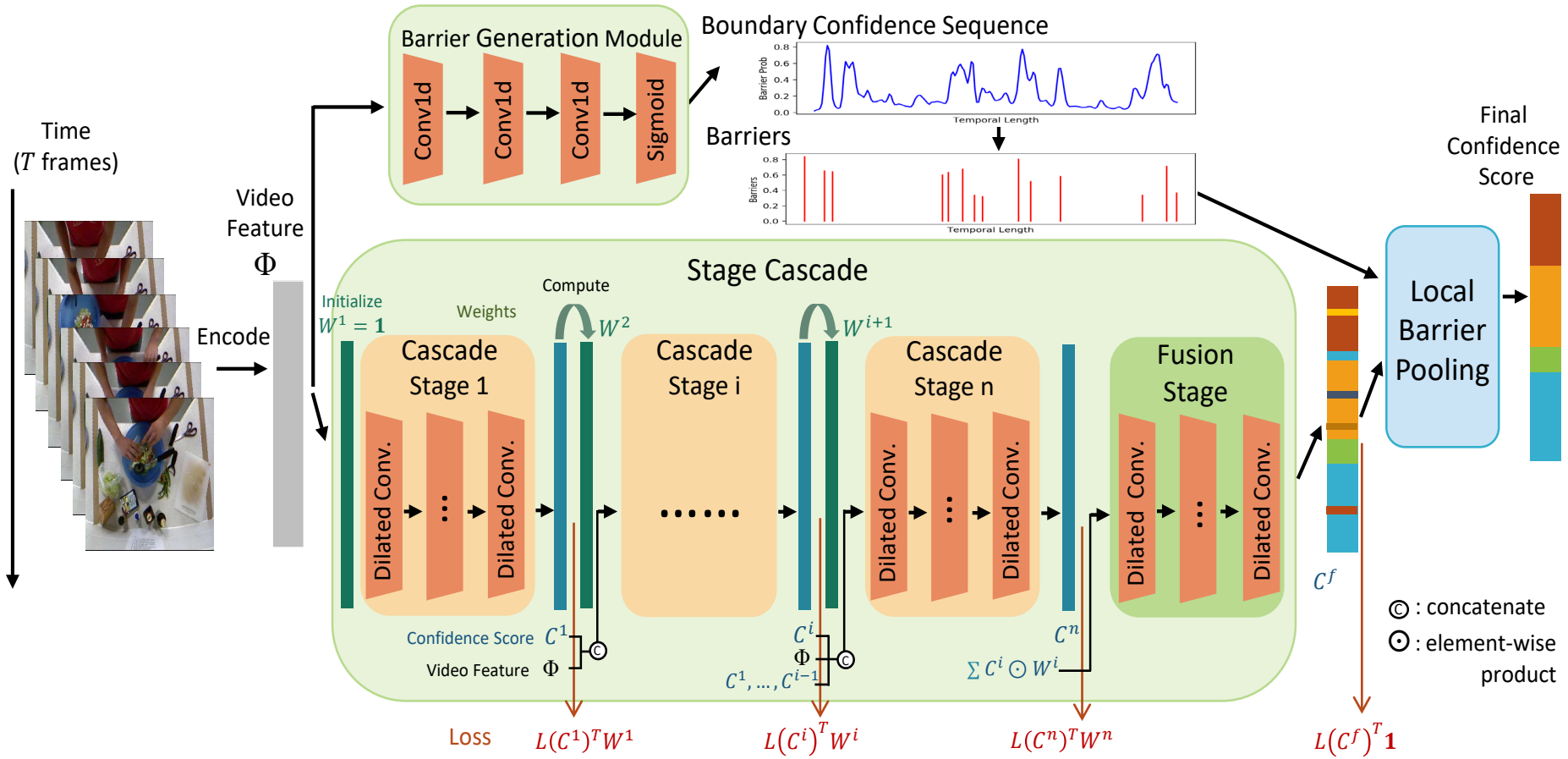


Fig.4: Overall framework

Evaluation: Stage Cascade

SC alone greatly improves frame-wise accuracy

Table 1: Comparison with baseline and BCN's variants on 50Salads (mid). The 1st and 2nd of each criterion are boldfaced and underlined respectively. (* reported in [3])

Methods	F1@{10,25,50}			Edit	Acc
MS-TCN*	76.3	74.0	64.5	67.9	80.7
MS-TCN w/ feature*	56.2	53.7	45.8	47.6	76.8
MS-TCN (5 stages)*	76.4	73.4	63.6	69.2	79.5
MS-TCN (12 layers)*	77.8	75.2	66.9	69.6	80.5
Stage Cascade	56.4	54.3	48.9	52.6	<u>83.4</u>
MS-TCN w/ LBP	78.3	75.9	66.1	68.1	81.5
MS-TCN w/ attention&LBP	<u>78.9</u>	<u>77.2</u>	<u>68.5</u>	<u>71.3</u>	82.7
BCN (SC w/ LBP)	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**

Evaluation: Stage Cascade

Detailed visualization shows our SC mainly improves accuracy in boundary regions.

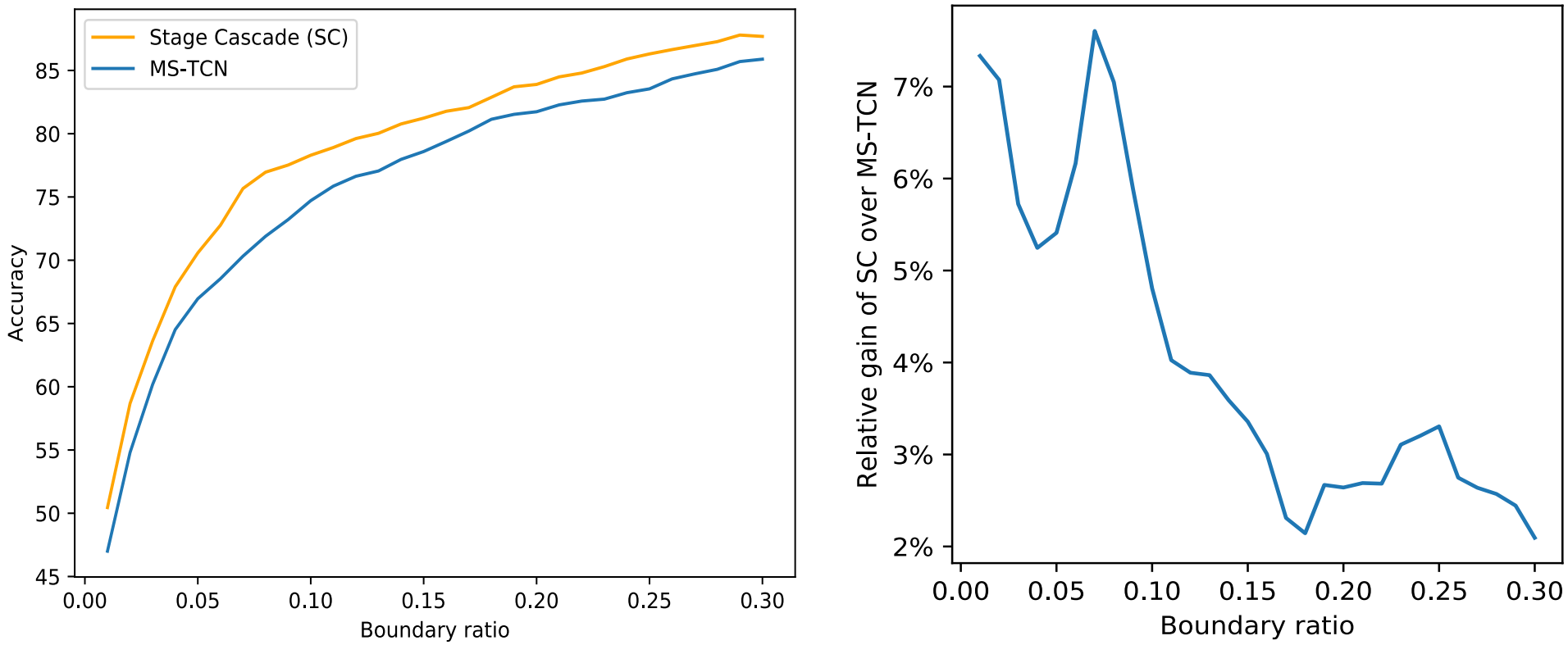


Fig.5: Stage Cascade's accuracy gain on 50Salads dataset.

Evaluation: Local Barrier Pooling

Table 1: Comparison with baseline and BCN's variants on 50Salads (mid). The 1st and 2nd of each criterion are boldfaced and underlined respectively. (* reported in [3])

Methods	F1@{10,25,50}			Edit	Acc
MS-TCN*	76.3	74.0	64.5	67.9	80.7
MS-TCN w/ feature*	56.2	53.7	45.8	47.6	76.8
MS-TCN (5 stages)*	76.4	73.4	63.6	69.2	79.5
MS-TCN (12 layers)*	77.8	75.2	66.9	69.6	80.5
Stage Cascade	56.4	54.3	48.9	52.6	<u>83.4</u>
MS-TCN w/ LBP	78.3	75.9	66.1	68.1	81.5
MS-TCN w/ attention&LBP	<u>78.9</u>	<u>77.2</u>	<u>68.5</u>	<u>71.3</u>	82.7
BCN (SC w/ LBP)	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**

LBP alone improves all metrics, especially F1 and edit score.

Table 2: LBP and two heuristic smoothing operators on 50Salads (mid).

Smoothing Operators	F1@{10,25,50}			Edit	Acc
Average (all barriers set as '0')	80.1	77.3	69.1	72.7	82.4
Gaussian-like (all barriers set as '1')	77.0	74.6	64.9	68.7	82.5
LBP (barriers from BGM)	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**

LBP outperforms its two special cases in fair comparison

Table 3: Comparison of LBP and λ in MS-TCN on 50Salads (mid)

50 Salads (mid)	F1@{10,25,50}			Edit	Acc
MS-TCN ($\lambda = 0.05$)*	74.1	71.7	62.4	66.6	80.0
MS-TCN ($\lambda = 0.15$)*	76.3	74.0	64.5	67.9	80.7
MS-TCN ($\lambda = 0.25$)*	74.7	72.4	63.7	**68.1**	78.9
MS-TCN w/ LBP	**78.3**	**75.9**	**66.1**	**68.1**	**81.5**

Additional loss function in previous state-of-the-art method can not match LBP's performance

Evaluation: Comparison with SOTA

Our method (BCN) outperforms previous state-of-the-art methods in 50Salads, GTEA, Breakfast datasets.

Table 5: Comparison with the state-of-the-art on 50Salads, GTEA and Breakfast dataset. (* uses multi-modal data, † obtained from [2])

50Salads (mid)	F1@{10,25,50}			Edit	Acc
Spatial CNN [14]	32.3	27.1	18.9	24.8	54.9
IDT+LM [22]	44.4	38.9	27.8	45.8	48.7
Dilated TCN [13]	52.2	47.6	37.4	43.1	59.3
ST-CNN [14]	55.9	49.6	37.1	45.9	59.4
Bi-LSTM [25]	62.6	58.3	47.0	55.6	55.7
ED-TCN [13]	68.0	63.9	52.6	59.8	64.7
TDRN [16]	72.9	68.5	57.2	66.0	68.1
MS-TCN [3]	76.3	74.0	64.5	67.9	80.7
Coupled GAN [6]*	80.1	78.7	71.1	76.9	74.5
BCN	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**

GTEA	F1@{10,25,50}			Edit	Acc
Bi-LSTM [25]	66.5	59.0	43.6	-	55.5
ED-TCN [13]	72.2	69.3	56.0	-	64.0
TDRN [16]	79.2	74.4	62.7	74.1	70.1
MS-TCN [3]	85.8	83.4	69.8	79.0	76.3
Coupled GAN [6]*	80.1	77.9	69.1	72.8	78.5
BCN	**88.5**	**87.1**	**77.3**	**84.4**	**79.8**

Breakfast	F1@{10,25,50}			Edit	Acc
ED-TCN [13]†	-	-	-	-	43.3
TCFPN [2]	-	-	-	-	52.0
HTK (64) [12]	-	-	-	-	56.3
GRU [23]†	-	-	-	-	60.6
MS-TCN (I3D) [3]	52.6	48.1	37.9	61.7	66.3
BCN	**68.7**	**65.5**	**55.0**	**66.2**	**70.4**

Ablation Study

Although some introduced dataset-specific hyper-parameters, our model greatly outperforms previous SOTA on other two datasets using the hyper-parameters obtained in 50Salads.

Table 3: Study on probability thresholds for stage cascade on 50Salads (mid).

Threshold	F1@{10,25,50}			Edit	Acc
0.5	81.6	79.7	71.5	74.2	83.0
0.6	82.1	80.9	71.9	74.1	83.3
0.7	81.9	80.5	72.4	74.0	83.9
0.8	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**
0.9	80.6	79.1	69.7	73.3	82.9

Table 4: Study on the the number of Cascade Stages on 50Salads dataset.

50Salads (mid)	F1@{10,25,50}			Edit	Acc
MS-TCN (3 stages)	71.5	68.6	61.1	64.0	78.6
MS-TCN (4 stages)	76.3	74.0	64.5	67.9	80.7
BCN (2 cascade stages)	81.2	79.2	71.3	73.5	83.6
BCN (3 cascade stages)	**82.3**	**81.3**	**74.0**	**74.3**	**84.4**
BCN (4 cascade stages)	80.7	78.9	71.5	72.9	83.5

Our method with less stages (less capacity and computation cost) outperforms baseline's performance

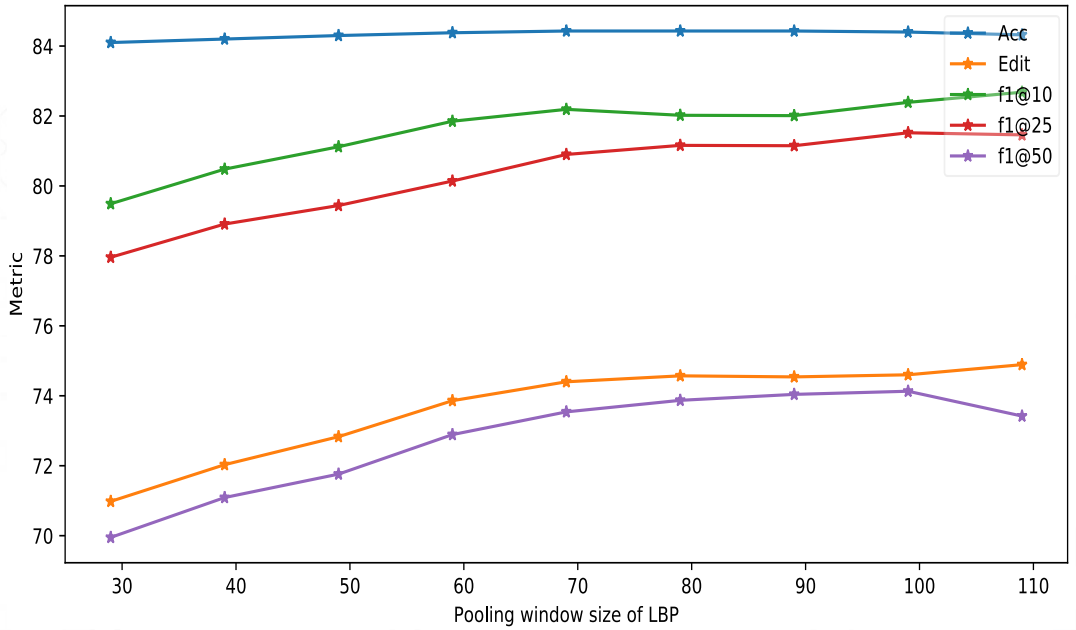


Fig.6: Ablation study of LBP's pooling window size on 50Salads (mid).

Visualization: 50Salads

4 rows:

- Frame-wise entropy of SC vs. baseline
- Action segmentation results of groundtruth, BCN, SC and baseline
- Performance improvement (red) and drop (green) by SC, and baseline's error (purple)
- The cascade stage among stage 1,2 or 3 which dominates the weights is in blue.

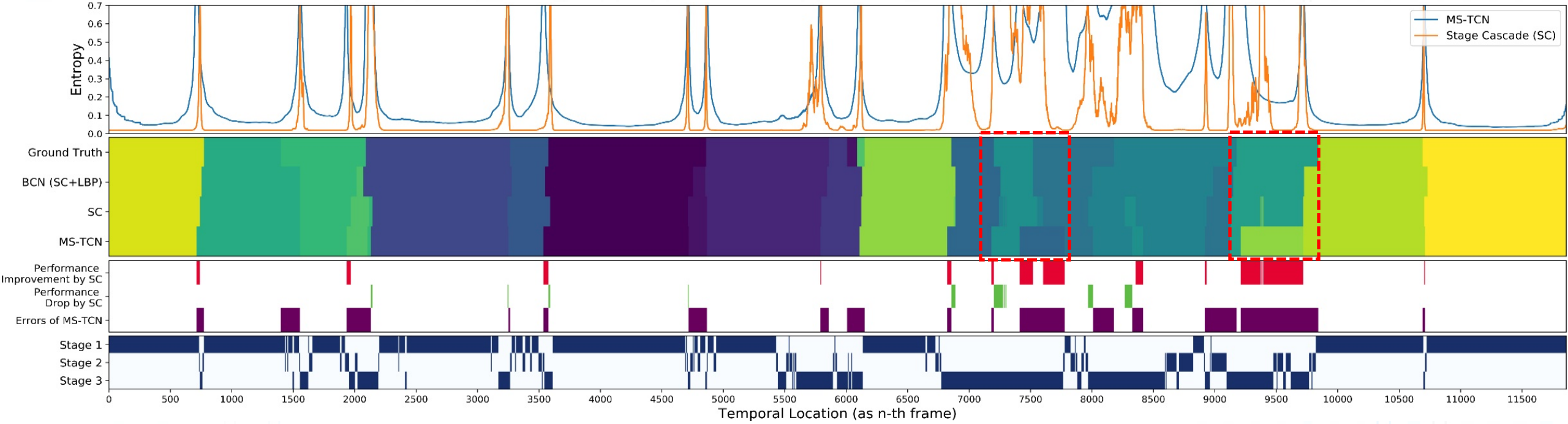


Fig.7: Qualitative result of 50Salads

Visualization: Breakfast

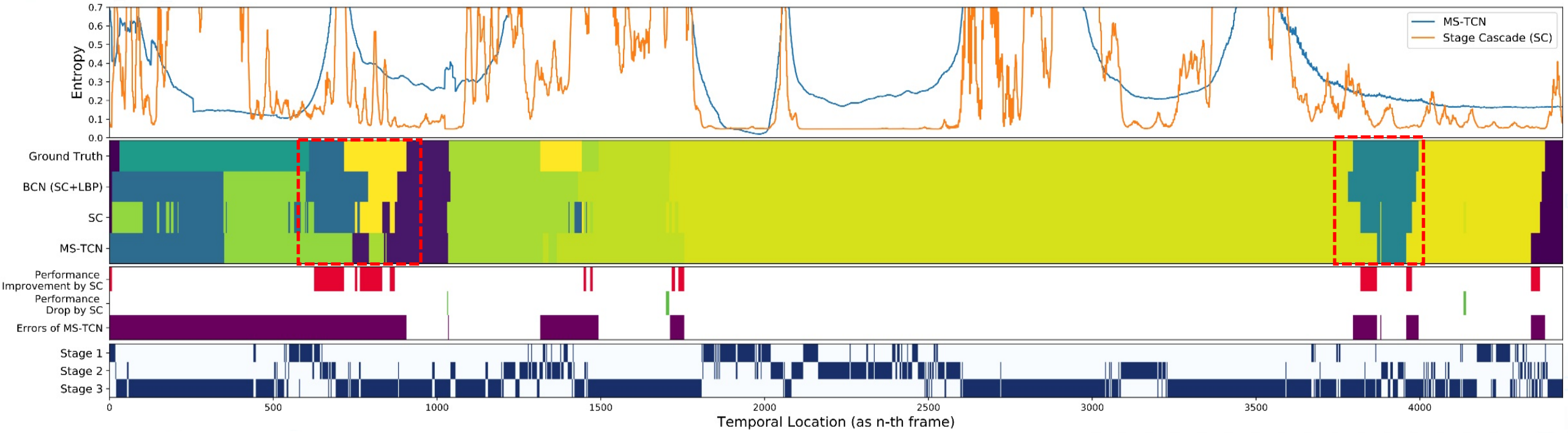


Fig.8: Qualitative result of Breakfast

Visualization: GTEA

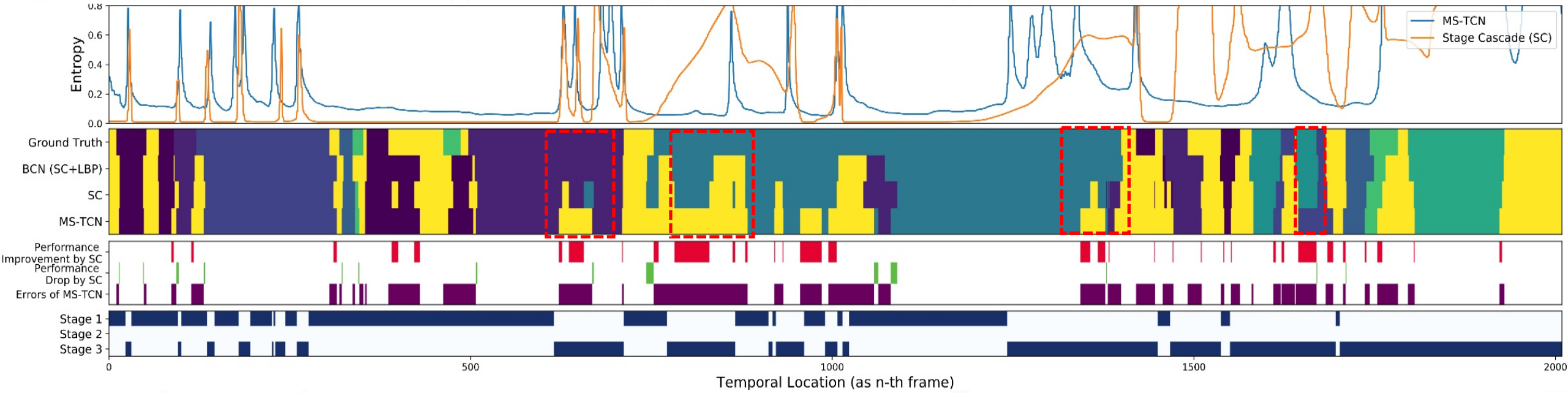


Fig.9: Qualitative result of GTEA

Thank you for your attention!

Code will be available at https://github.com/MCG-NJU/BCN.