Stacked encoder–decoder transformer with boundary smoothing for action segmentation

Gyeong-hyeon Kim and Eunwoo Kim✉
School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea
✉Email: eunwoo@cau.ac.kr

In this work, a new stacked encoder–decoder transformer (SEDT) model is proposed for action segmentation. SEDT is composed of a series of encoder–decoder modules, each of which consists of an encoder with self-attention layers and a decoder with cross-attention layers. By adding an encoder with self-attention before every decoder, it preserves local information along with global information. The proposed encoder–decoder pair also prevents the accumulation of errors that occur when features are propagated through decoders. Moreover, the approach performs boundary smoothing in order to handle ambiguous action boundaries. Experimental results for two popular benchmark datasets, "GTEA" and "50 Salads", show that the proposed model is more effective in performance than existing temporal convolutional network based models and the attention-based model, ASFormer.

Introduction: Video understanding is required in many diverse fields supporting human life, such as surveillance systems, robotics, medical care, and silver care. Accordingly, many efforts to understand the video have been attempted. In this paper, we focus on a temporal action segmentation task. Temporal action segmentation is to understand human activity and how long it has lasted in a long untrimmed video. In other words, it is a dense classification of all frames of a video. Action segmentation is not just classification for each video frame but context understanding across video frames. A model should be able to consider the global context of the video as well as the local context between neighbour frames.

In previous works [1–3], the temporal convolutional network (TCN) was one of the dominant approaches for action segmentation. It is one of the 1D convolutional networks along with the time axis. TCN can aggregate local context between neighbour frames with a fixed kernel size. Based on this, TCN can get a global context of a video by aggregating local contexts through stacked layers. Note that the global context is gradually strengthened, but the local context fades as the TCN layer deepens. To solve this problem, ASFormer [4] firstly applied the self-attention mechanism, which was popularly applied to NLP tasks, on action segmentation. Unlike convolutional layers, self-attention can get global context at once since it calculates the relationship between all features. But, ASFormer still has the same problem of fading local context as other TCN-based models [1–3].

To cover this, we propose a new stacked encoder–decoder transformer model (SEDT) for action segmentation. An encoder is added before each decoder on the transformer-based framework [4] to form an encoder–decoder pattern [5, 6]. Additional encoder returns a new initial prediction from the decoder output feature, preventing the accumulation of errors that may appear from successive decoders. In addition, we introduce a new boundary smoothing strategy that lessens the ambiguity of the action class near the action boundary. Our framework yields state-of-the-art results on two challenging datasets: 50 Salads [7] and GTEA [8]. On 50 Salads, our framework records higher F1 scores of 4.8, 5.3, and 5.1 for the thresholds 0.1, 0.25, and 0.5, respectively, than the state-of-the-art transformer-based model [4]. On GTEA, our method also outperforms existing approaches for all metrics.

Proposed architecture: In this section, we present our stacked encoder–decoder transformer (SEDT) for temporal action segmentation as shown in Figure 1. The proposed SEDT model is composed of stacked N encoder–decoder modules, each of which consists of an encoder and a decoder. This encoder–decoder module pattern [5, 6] can prevent error accumulation caused by successive decoders from ASFormer [4]. The encoder with stacked encoder blocks predicts initial prediction with the self-attention mechanism, and the decoder with stacked decoder blocks refines the prediction of the encoded feature from the encoder. The decoder gets class prediction probability by the encoder as a new input,

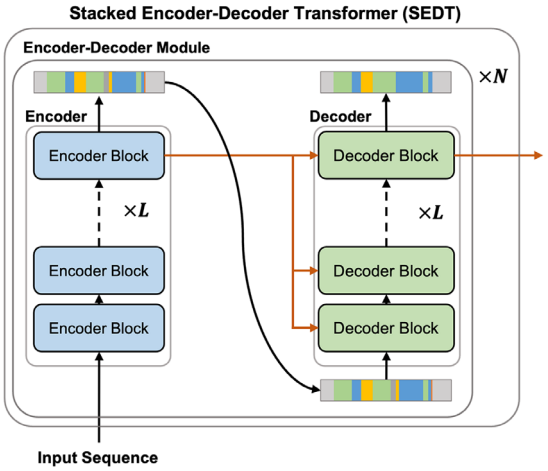
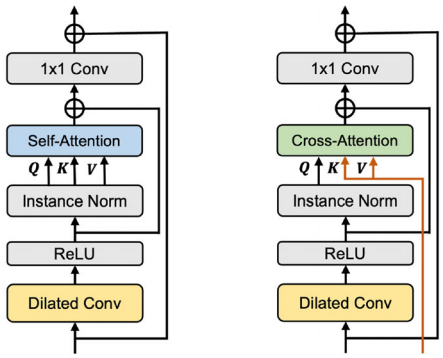


Fig. 1 *An overview of the stacked encoder–decoder transformer model (SEDT), which consists of N encoder–decoder modules. Each encoder–decoder module has an encoder and a decoder, each of which consists of L encoder blocks and L decoder blocks, respectively*



(a) *Encoder block* (b) *Decoder block*

Fig. 2 *Illustration of the (a) encoder and (b) decoder block*

and the encoder–decoder module conveys the refined feature to the next encoder–decoder module.

Encoder: An encoder consists of several encoder blocks shown in Figure 1. It consists of two sub-layers, a feed-forward layer and a self-attention layer, as shown in Figure 2a. We use a dilated temporal convolutional layer as a feed-forward layer to overcome the lack of the large training dataset instead of a point-wise fully connected layer originally used in transformer [9]. The dilation of the feed-forward layer is doubled between layers i and $i+1$ (e.g. $2^{(i-1)}$, $i = 1, 2, \ldots$). The linear transformation with dilation rate is followed by ReLU activation:

$$\begin{aligned} X_l' &= \mathrm{FF}(X_l), \\ \mathrm{FF}(x) &= \mathrm{ReLU}(xW + b), \end{aligned} \tag{1}$$

where FF consists of a dilated convolutional layer and an activation. X_l and X_l' are input features of the lth encoder block and output features of FF, respectively. FF is followed by instance normalization [10]:

$$X_l'' = \mathrm{InstanceNorm}\,(X_l'). \tag{2}$$

The attention mechanism for the self-attention layer is as follow:

$$\begin{aligned} Q = X_l''W_q, K = X_l''W_k, V = X_l''W_v \\ \mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(QK^T \backslash \sqrt{d_k})V, \end{aligned} \tag{3}$$

where W_q, W_k, W_v are the learnable query, key and value matrices, respectively, and $\sqrt{d_k}$ is a scaling factor. The self-attention layer gets a query, a key, and a value through the linear projections of the feature from the feed-forward layer. It aggregates the global context by calculating the relationship between all frames in the video with

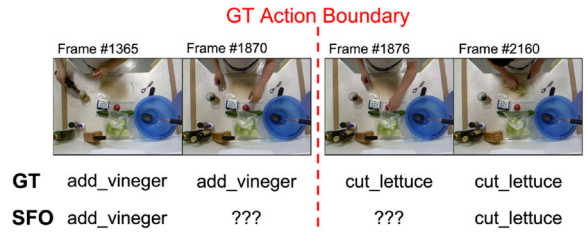


Fig. 3 *An example of ambiguous frames near action boundary on 50 Salads*

attention in Equation (3). We use single-head self-attention instead of multi-head self-attention following ASFormer [4]. ASFormer experimentally shows that the performance difference between single-head and multi-head self-attention is not significant. Residual connection is applied before two sub-layers:

$$X_{l+1} = \sigma(\text{Attention}(Q, K, V) + X_l'') + X_l, \tag{4}$$

where σ is a 1×1 convolutional layer to adjust the number of channels for the next encoder block. X_{l+1} is the output of the lth block and the input of the $(l+1)$th block. We do not use positional encoding following ASFormer [4] since the temporal convolutional layer used in the feed-forward layer can model the relative position of frames.

Decoder: The structure of the decoder block is the same as the encoder block except for one sub-layer. As shown in Figure 2b, The type of attention mechanism is the only difference between the encoder and the decoder blocks. The cross-attention layer gets a query from the previous feed forward layer, and a key and a value from the previous encoder:

$$Q = X_l'' W_q, K = EW_k, V = EW_v, \tag{5}$$

where E is the encoded feature from the previous encoder. The same attention operation in Equation (3) is applied. In the case of the first decoder block on the decoder, it gets a query from the class probability of the encoder output.

Boundary smoothing (BS): The action segmentation model takes input videos as a sequence of frames and returns an action class for each frame. Since an action takes place across frames, an action consists of a sequence of several frames. It is ambiguous to label a single action on a frame where an action transition occurs near the action boundary. Note that an action transition is a move to the next action. We observed that the action boundaries are ambiguous on one-hot encoded ground-truth (GT) labels. As shown in Figure 3, frames #1870 and #1876 are hard to classify an action with the single frame observation (SFO). More specifically, many frames are associated with one or more actions, making it difficult to distinguish some actions among frames near the action boundary due to overlapping actions. Considering this, we present an approach where the original GT class probability gradually decreases as the frame approaches the action boundary.

Inspired by ref. [11], the cosine function is adopted to smoothly decrease probability as

$$\text{BS}(x_{\text{pos}}, T) = a_{\min} + \frac{1}{2}(a_{\max} - a_{\min})\left(1 + \cos\frac{x_{\text{pos}}}{T}\pi\right), \tag{6}$$

where T is the half-length of the action segment including the input frame. $a_{\max}$ and $a_{\min}$ are possible maximum and minimum confidence of the original action class, respectively. x_{pos} is the relative frame position from the centre frame of the action segment. For example, x_{pos} is in range of $-T$ to T. The decreased GT class probability by boundary smoothing in each frame is given to the neighbour action segment class. But, the action segment can have two (left and right) neighbour action segments. We divide the action segment into two sub-segments based on the centre frame of the action segment. Now, each of the sub-segments can have only one neighbour segment. Then, the decreased probability by boundary smoothing is assigned to the neighbour segment class for each sub-segment. In other words, the farther away from the centre frame within the action segment, the more noise is given to the neighbour action segment class. Boundary-smoothed GT labels are used to train the model instead of the original GT labels.

Table 1. The statistics of the action segmentation datasets

Dataset	#class	#video	Splits	Viewpoint	Scene
GTEA [8]	11	28	4	Egocentric	Activity in kitchen
50 Salads [7]	19	50	5	Top-view	Preparing salads

Loss function: Following the work [1], the combination of frame-wise classification loss $\mathcal{L}_{\text{cls}}$ and truncated mean square error loss $\mathcal{L}_{\text{t-mse}}$ was used for the loss function. The classification loss is the cross-entropy loss for each frame, and the truncated mean square error loss is a smooth loss that prevents drastic change over class probability between consecutive frames. The classification loss $\mathcal{L}_{\text{cls}}$ and the smooth loss $\mathcal{L}_{\text{t-mse}}$ are

$$\mathcal{L}_{\text{cls}} = \frac{1}{T}\sum_t -\log(y_{t,c}), \tag{7}$$

and

$$\mathcal{L}_{\text{t-mse}} = \frac{1}{TC}\sum_{t,c}(\log y_{t,c} - \log y_{t-1,c})^2, \tag{8}$$

where $y_{t,c}$ is the predicted probability for the ground truth label c at time t. T and C are the number of video frames and action classes, respectively. We use a combination of these two losses with a balance weight λ which was set to 0.25 in our experiments. The total loss $\mathcal{L}$ is

$$\mathcal{L} = \sum_s \mathcal{L}_{\text{cls}}^s + \lambda \mathcal{L}_{\text{t-mse}}^s, \tag{9}$$

where s is the index of the encoders and decoders in the model. For model training, we minimize the sum of losses collected from all encoders and decoders.

Dataset: To evaluate proposed architecture, we use two challenging datasets: Georgia Tech Egocentric Activities (GTEA) [8] and 50 Salads [7]. The statistics for the datasets are listed in Table 1. We used the I3D [12] feature, which was extracted from all video frames with the dimension of 2048. GTEA is originally 15 fps (frames per second) videos, and 50 Salads is 30 fps videos. We downsampled 50 Salads I3D features to achieve 15 fps to be consistent with GTEA for model training. When evaluating the model performance on 50 Salads, we duplicate all frame results to fit 30 fps to assess with the original ground truth class label, following the practice in ref. [1].

Experimental setup: Our model consists of three encoder–decoder modules. Each encoder–decoder module consists of an encoder and a decoder, consisting of 10 encoder blocks and 10 decoder blocks, respectively. The model was trained 120 epochs with the Adam optimizer. Channel dropout [13] is used with the rate of 0.5 and 0.3 for GTEA [8] and 50 Salads [7], respectively. The batch size was 1 for all experiments. The channel dimension of all sub-layers was 64. For boundary smoothing, we set $a_{\max}$ and $a_{\min}$ to 1.0 and 0.9, respectively. We followed previous works [1–4] to evaluate our methods: frame-wise accuracy (Acc), segmental edit distance (Edit), and segmental F1 score at temporal intersection over union with thresholds 0.1, 0.25, 0.5 (F1@10, F1@25, F1@50). These three metrics are commonly used for action segmentation.

Quantitative results: We compared proposed methods to state-of-the-art frameworks based on temporal convolutional networks (TCN) and attention mechanisms (Attention). As shown in Table 2, Our model with boundary smoothing achieves state-of-the-art results except for accuracy on 50 Salads. SEDT outperforms existing TCN and attention-based methods for all metrics. Our method with boundary smoothing has higher F1 scores of 4.8, 5.3, 5.1 for the thresholds 0.1, 0.25, 0.5 (F1@10, 25, 50), respectively, compared to ASFormer [4]. As for segmental edit distance, SEDT also shows 4.9 higher performance than ASFormer. Ours without boundary smoothing (SEDT†), the method

*Table 2. Comparison with existing state-of-the-art TCN and attention-based methods on 50 Salads. SEDT is the proposed approach with boundary smoothing. SEDT† is that without boundary smoothing. Metric with **bold** shows the best record, and the underlined gives the second best*

Method		F1@{10, 25, 50}			Edit	Acc
TCN	MS-TCN [1]	76.3	74.0	64.5	67.9	80.7
	ASRF [2]	84.9	83.5	77.3	79.3	84.5
	C2F-TCN [3]	84.3	81.8	72.6	76.4	84.9
Attention	ASFormer [4]	85.1	83.4	76.0	79.6	85.6
	SEDT† (ours)	87.0	85.8	79.0	81.1	**86.7**
	SEDT (ours)	**89.9**	**88.7**	**81.1**	**84.7**	86.5

Table 3. Comparison with existing state-of-the-art methods on GTEA

Method		F1@{10, 25, 50}			Edit	Acc
TCN	MS-TCN [1]	85.8	83.4	69.8	79.0	76.3
	ASRF [2]	89.4	87.8	79.8	83.7	77.3
	C2F-TCN [3]	90.3	88.8	77.7	86.4	80.8
Attention	ASFormer [4]	90.1	88.8	79.2	84.6	79.7
	SEDT† (ours)	92.2	90.5	82.7	89.9	80.8
	SEDT (ours)	**93.7**	**92.4**	**84.0**	**91.3**	**81.3**

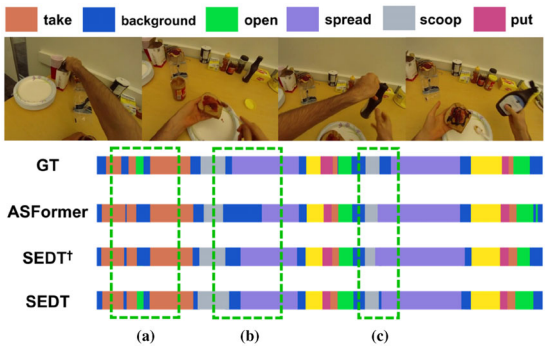


Fig. 4 *Qualitative results of the compared methods on GTEA*

records the second best for all metrics except accuracy. It shows that SEDT is better for learning the context of video than TCN-based and attention-based methods. It also shows that the boundary smoothing strategy enables additional performance gains by lessening the ambiguity near the action boundary. On GTEA, our model outperforms TCN-based and attention-based models for all metrics, as shown in Table 3. SEDT without boundary smoothing also performs better than TCN and attention-based methods.

Qualitative results: We conducted a qualitative analysis of the proposal with ASFormer [4]. Figure 4 visualizes the results of action segmentation on GTEA. Our method detects action segments more precisely than ASFormer, as shown in Figure 4b. It shows that the proposed module is effective in aggregating local and global contexts in the video. Compared to ASFormer and SEDT†, SEDT with boundary smoothing is better for detecting small action segments such as "open" and "background" as shown in Figure 4a,c. It shows that the proposed boundary smoothing strategy effectively detects short action segments.

Conclusion: We have presented a new SEDT by adding an encoder prior to every decoder. By stacking self-attention and cross-attention alternatively, SEDT preserves local information with global information and prevents error accumulation caused by propagating only decoders. Moreover, we have proposed boundary smoothing to handle ambiguous action boundaries, showing additional performance improvements. The proposed framework yields state-of-the-art performance in temporal action segmentation on two challenging datasets. However, SEDT has several limitations, such as fading local information inside the encoder and high computational costs to calculate attention scores between all frames. These limitations will be addressed in future studies.

Author Contributions: Gyeong-hyeon Kim: Conceptualization, investigation, methodology, validation, visualization, writing - original draft. Eunwoo Kim: Resources, supervision, writing - review and editing.

Acknowledgements: This work was supported in part by the Chung-Ang University Graduate Research Scholarship in 2021, in part by Institute of Information & communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program (Chung-Ang University)), and in part by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE, Ministry of Trade, Industry and Energy) (P0012724, The Competency Development Program for Industry Specialist).

Conflict of Interest: The authors declare no conflict of interest.

Data availability statement: Data sharing not applicable—no new data generated, or the article describes entirely theoretical research

Received: *22 September 2022* Accepted: *7 November 2022*
doi: 10.1049/ell2.12678

References

1 Farha, Y.A., Gall, J.: MS-TCN: multi-stage temporal convolutional network for action segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3575–3584. IEEE, Piscataway, NJ (2019)
2 Ishikawa, Y., Kasai, S., Aoki, Y., Kataoka, H.: Alleviating oversegmentation errors by detecting action boundaries. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2322–2331. IEEE, Piscataway, NJ (2021)
3 Singhania, D., Rahaman, R., Yao, A.: Coarse to fine multi-resolution temporal convolutional network. arXiv:210510859 (2021)
4 Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. arXiv:211008568 (2021)
5 Zhang, R., Shu, X., Yan, R., Zhang, J., Song, Y.: Skip-attention encoder–decoder framework for human motion prediction. *Multimedia Syst.* **28**(2), 413–422 (2022)
6 Shu, X., Zhang, L., Qi, G.J., Liu, W., Tang, J.: Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(6), 3300–3315 (2021)
7 Stein, S., McKenna, S.J.: Recognising complex activities with histograms of relative tracklets. *Computer Vision and Image Understanding* **154**, 82–93 (2017)
8 Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: 2011 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3281–3288. IEEE, Piscataway, NJ (2011)
9 Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. In: 31st Conference on Neural Information Processing Systems, pp. 1–11. Curran Associates, Red Hook, NY (2017)
10 Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization. arXiv:160708022 (2016)
11 Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. arXiv:160803983 (2016)
12 Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6299–6308. IEEE, Piscataway, NJ (2017)
13 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)