Supplementary Materials: Progress-Aware Online Action Segmentation for Egocentric Procedural Task Videos

Yuhan Shen
Northeastern University
shen.yuh@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

In this supplementary material, we include

- predefined task graphs on the EgoPER dataset;
- additional implementation details about our causal action segmentation models;
- additional experimental results on Assembly 101 dataset [3];
- additional experimental results on offline action segmentation.

1. Task Graphs on EgoPER

The EgoPER dataset [2] consists of videos of five tasks/recipes, including ‘making coffee’, ‘making pinwheel’, ‘making tea’, ‘making oatmeal’, and ‘making quesadilla’. Each task is associated with a task graph, as illustrated in Figure 1. The task graphs encode all possible ways that the recipe could be made, and the participants followed one of these possible ways to perform the task. Please refer to [2] for more details on the dataset statistics.

2. Implementation Details

We use an MSTCN [1] architecture with four stages, and each stage contains ten dilated convolution layers, where the dilation factor is doubled at each layer and dropout is used after each layer. The number of convolution filters is 64, and the filter size is 3. For ASFormer [5], we use one encoder and three decoders, while each encoder and decoder contain nine blocks. The dimension of the first fully connected layer is set to 64 in both encoder and decoders. For APP modules, we use a uni-directional GRU with hidden dimension of 64 followed by a fully-connected layer to output the progress estimations. We optimize the model using Adam optimizer with a learning rate of 0.0005. We set the loss weights as $\lambda_{\text{smo}} = 0.15, \lambda_{\text{prog}} = 1, \lambda_{\text{graph}} = 0.1$.

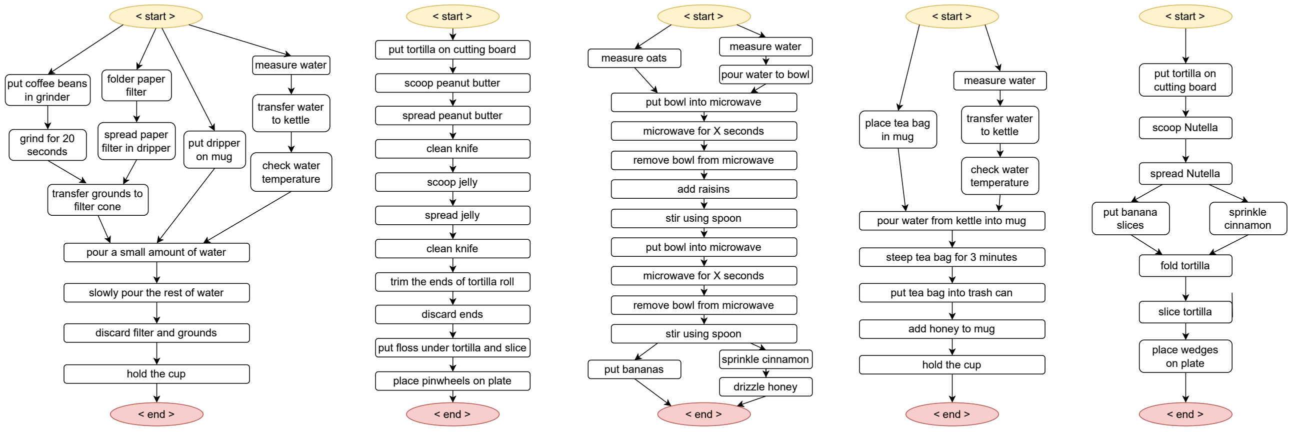
3. Additional Experimental Results

Results on Assembly 101. Assembly 101 is a large *procedural* dataset consisting of both egocentric and exocentric videos. However, the objects (toys) in Assembly 101 are small, making it challenging to capture progress, and the egocentric videos are grey-scale, therefore we did not use it in our main paper. To ensure the completeness of our study, we include a comparison using the egocentric views from Assembly 101 in Table 1. Following the official repository, we use C2F-TCN [4] as the backbone. We observe that the integration of our proposed components (CAS, APP, and the task graph) yields improvements even on this challenging dataset For example, in terms of F1@0.5, in terms of the F1@0.5 metric, the CAS component alone accounts for a 2.6% increase, while APP further improves it by 0.6%. Moreover, the combination of all three components boosts the performance from 6.5% to 12.2%. Nevertheless, it is important to acknowledge that these performance gains are less pronounced when compared to those achieved on other datasets.

Method	Acc	Edit	F1@{0.1,0.25,0.5}		
Base (offline)	34.8	29.2	28.7	24.4	17.5
Base (online)	20.7	14.8	14.4	11.2	6.5
CAS	21.6	18.0	19.4	15.7	9.1
CAS+APP	22.1	18.4	20.3	16.4	9.7
CAS+APP+TG	**24.5**	**21.4**	**21.3**	**18.2**	**12.2**

Table 1. Results on Assembly 101.

Offline Action Segmentation. Due to space constraints, we only report the results of offline action segmentation on EgoProceL, so we include the results on all the three datasets in Table 2. For GTEA, we show the results of MSTCN reported in the original paper [1] as well as the results we reproduced (our run) as we achieve better performance than what was reported in the paper. We notice that the trends on all three datasets are similar. While APP and TG can enhance the segmentation results in some cases, the improvement is marginal compared with their impacts in online scenarios. This is because the offline model is able to consider the future frames to correct the prediction errors



(a) coffee (b) pinwheel (c) oatmeal (d) tea (e) quesadilla

Figure 1. Task graphs of the five procedural activities on EgoPER.

Dataset	Method	Acc	Edit	F1@{0.1,0.25,0.5}		
GTEA	MSTCN	76.3	79.0	85.8	83.4	69.8
	MSTCN (our run)	**80.3**	81.4	87.2	83.8	72.6
	MSTCN+APP	80.2	**82.7**	**87.3**	84.4	**72.9**
	MSTCN+APP+TG	80.1	81.8	**87.3**	**85.3**	72.2
EgoProceL	MSTCN	69.2	56.9	58.9	55.8	45.9
	MSTCN+APP	70.3	56.6	60.6	56.9	**46.8**
	MSTCN+APP+TG	**71.1**	**60.4**	**63.3**	**59.3**	46.1
EgoPER	MSTCN	**83.0**	**85.9**	**88.9**	**87.4**	77.3
	MSTCN+APP	82.9	85.6	88.5	87.1	78.0
	MSTCN+APP+TG	82.8	85.4	88.5	87.2	**78.1**

Table 2. Offline action segmentation performance on three datasets.

without the need of modeling the action progress or using task graph. This observation underscores the effectiveness of progress estimation and task graph in the context of online action segmentation, where their contributions are particularly significant.

References

[1] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019.

[2] S. Lee, Z. Lu, Z. Zhang, M. Hoai, and E. Elhamifar. Error detection in egocentric procedural task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[3] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022.

[4] Dipika Singhania, Rahul Rahaman, and Angela Yao. Coarse to fine multi-resolution temporal convolutional network. *CoRR*, abs/2105.10859, 2021.

[5] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *The British Machine Vision Conference (BMVC)*, 2021.