# Supplementary Materials: Semi-Weakly-Supervised Learning of Complex Actions from Instructional Task Videos

Yuhan Shen
Northeastern University
shen.yuh@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

## 1. Backward Propagation for SRE

To obtain $\nabla_{\boldsymbol{Q}} \text{SRE}(\boldsymbol{G}, \boldsymbol{Q})$, we first calculate the derivatives of SRE Loss $J$ w.r.t. the entries of the cost matrix $\boldsymbol{\Delta}(\boldsymbol{G}, \boldsymbol{Q})$, which can be computed by chain rule:

$$\begin{aligned}\frac{\partial J}{\partial \delta_{i,j}} &= \sum_{i',j'} \frac{\partial J}{\partial e_{i',j'}} \frac{\partial e_{i',j'}}{\partial \delta_{i,j}} \\ &= \frac{\partial J}{\partial e_{i+1,j+1}} \frac{\partial e_{i+1,j+1}}{\partial \delta_{i,j}} + \frac{\partial J}{\partial e_{i+1,j+2}} \frac{\partial e_{i+1,j+2}}{\partial \delta_{i,j}} \\ &+ \frac{\partial J}{\partial e_{i+2,j+1}} \frac{\partial e_{i+2,j+1}}{\partial \delta_{i,j}}.\end{aligned} \tag{1}$$

Recall the definition of cumulative cost matrix $[e_{i,j}]$:

$$e_{i,j} = \min_\beta \begin{cases} e_{i-1,j} + c_D, \\ e_{i,j-1} + c_I, \\ e_{i-1,j-1} + \delta_{i-1,j-1}, \\ e_{i-2,j-2} + \delta_{i-2,j-1} + \delta_{i-1,j-2} + c_T \ (\forall i,j \geq 3). \end{cases} \tag{2}$$

From (2), we can get the following equations:

$$\begin{aligned}\frac{\partial e_{i+1,j+1}}{\partial \delta_{i,j}} &= \frac{\partial e_{i+1,j+1}}{\partial e_{i,j}}, \\ \frac{\partial e_{i+1,j+2}}{\partial \delta_{i,j}} &= \frac{\partial e_{i+1,j+2}}{\partial e_{i-1,j}}, \\ \frac{\partial e_{i+2,j+1}}{\partial \delta_{i,j}} &= \frac{\partial e_{i+2,j+1}}{\partial e_{i,j-1}}.\end{aligned} \tag{3}$$

Let us define two intermediate variables:

$$r_{i,j} = \frac{\partial J}{\partial \delta_{i,j}}, h_{i,j} = \frac{\partial J}{\partial e_{i,j}}, \tag{4}$$

and two auxiliary variables:

$$a_{i,j} = \frac{\partial e_{i+1,j+1}}{\partial e_{i,j}}, b_{i,j} = \frac{\partial e_{i+2,j+2}}{\partial e_{i,j}}. \tag{5}$$

Combining (3)(4)(5) into (1), we get:

$$r_{i,j} = h_{i+1,j+1} \cdot a_{i,j} + h_{i+1,j+2} \cdot b_{i-1,j} + h_{i+2,j+1} \cdot b_{i,j-1}. \tag{6}$$

To calculate $h_{i,j}$, we use chain rule for $h_{i,j} = \frac{\partial J}{\partial e_{i,j}}$ from (2):

$$\begin{aligned}\frac{\partial J}{\partial e_{i,j}} &= \sum_{i',j'} \frac{\partial J}{\partial e_{i',j'}} \frac{\partial e_{i',j'}}{\partial e_{i,j}} \\ &= \frac{\partial J}{\partial e_{i+1,j}} \frac{\partial e_{i+1,j}}{\partial e_{i,j}} + \frac{\partial J}{\partial e_{i,j+1}} \frac{\partial e_{i,j+1}}{\partial e_{i,j}} \\ &+ \frac{\partial J}{\partial e_{i+1,j+1}} \frac{\partial e_{i+1,j+1}}{\partial \delta_{i,j}} + \frac{\partial J}{\partial e_{i+2,j+2}} \frac{\partial e_{i+2,j+2}}{\partial e_{i,j}}.\end{aligned} \tag{7}$$

To simplify the notation, let $u = \frac{\partial e_{i+1,j}}{\partial e_{i,j}}$ and $v = \frac{\partial e_{i,j+1}}{\partial e_{i,j}}$, then we can rewrite (7) as:

$$\begin{aligned}h_{i,j} &= h_{i+1,j} \cdot u + h_{i,j+1} \cdot v \\ &+ h_{i+1,j+1} \cdot a_{i,j} + h_{i+2,j+2} \cdot b_{i,j}.\end{aligned} \tag{8}$$

Hence, we can update $h_{i,j}$ recursively from the last item by computing $u, v, a_{i,j}, b_{i,j}$, which is derived from (2) as follows:

$$u = \frac{\partial e_{i+1,j}}{\partial e_{i,j}} = \phi_\beta(e_{i,j} + c_D, e_{i+1,j}), \tag{9}$$

where $\phi_\beta(x, y) = \exp\{-(x-y)/\beta\}$.

Similarly, we compute the value of the other items as:

$$v = \frac{\partial e_{i,j+1}}{\partial e_{i,j}} = \phi_\beta(e_{i,j} + c_I, e_{i,j+1}), \tag{10}$$

$$a_{i,j} = \frac{\partial e_{i+1,j+1}}{\partial e_{i,j}} = \phi_\beta(e_{i,j} + \delta_{i,j}, e_{i+1,j+1}), \tag{11}$$

$$b_{i,j} = \frac{\partial e_{i+2,j+2}}{\partial e_{i,j}} = \phi_\beta(e_{i,j} + \delta_{i,j+1} + \delta_{i+1,j} + c_T, e_{i+2,j+2}). \tag{12}$$

Hence, we summarize the update rule for $h_{i,j}, a_{i,j}, b_{i,j}$ as follows:

$$\begin{aligned} u &= \phi_\beta(e_{i,j} + c_D, e_{i+1,j}), \\ v &= \phi_\beta(e_{i,j} + c_I, e_{i,j+1}), \\ a_{i,j} &= \phi_\beta(e_{i,j} + \delta_{i,j}, e_{i+1,j+1}), \\ b_{i,j} &= \phi_\beta(e_{i,j} + \delta_{i,j+1} + \delta_{i+1,j} + c_T, e_{i+2,j+2}), \\ h_{i,j} &= h_{i+1,j} \cdot u + h_{i,j+1} \cdot v \\ &\quad + h_{i+1,j+1} \cdot a_{i,j} + h_{i+2,j+2} \cdot b_{i,j}, \end{aligned} \tag{13}$$

where $\phi_\beta(x, y) = \exp\{-(x - y)/\beta\}$.

## 2. Assumption Justification

One important assumption we make for semi-weakly supervised learning is that *the unlabeled videos have task labels*, which is a common assumption used in unsupervised action segmentation works [1, 2, 4]. Instructional videos demonstrate how to perform a task, thus, each video often has a task label. It is much easier to obtain the task label than the transcript of an instructional video. For example, the HowTo100M dataset [3] is collected by searching videos via a query "*how to + task name*" on YouTube, which naturally gives the task labels without extra annotation effort.

Besides, our proposed Soft Restricted Edit (SRE) loss is based on the assumption that *the transcripts of the videos from the same task often have small but nonzero distances*. To justify this assumption, we computed the average pairwise normalized edit distance of the transcripts of videos. If we constrain each pair of transcripts to be within the same task, the distance is 0.297 on Breakfast and 0.485 on CrossTask, while it is 0.672 and 0.666 respectively without any constraint. This shows that the transcripts of the same task have a small distance, and explains why our method improves more on Breakfast than CrossTask.

## 3. Additional Implementation Details

For MuCon, we train the model using weakly-labeled videos for 60 epochs and then add unlabeled videos for training. We generate pseudo-transcripts for unlabeled videos in three rounds (in epoch 80, 100, 120), each round adding 1/3 of the unlabeled videos to the weakly-labeled set. We train the model for 180 epochs in total.

For CDFL, we train the model using weakly-labeled videos for 2000 iterations, and then add unlabeled videos for training. We generate pseudo-transcripts for unlabeled videos in three rounds (in iteration 5000, 10000, 15000), each round adding 1/3 of the unlabeled videos to the weakly-labeled set. We train the model for 60000 iterations in total.

When training MuCon on *CrossTask* dataset, we modify the input dimension of the neural network from 2048 to 3200 to comply with the feature dimension. For other experiments, we keep the network to be the same as in the original works.

The smoothing hyperparameter $\beta$ is 0.01. The $\mathcal{L}_{sre}$ weight $\rho$ is 0.1. We set the costs of insertion, deletion and adjacent transposition as $c_I = c_D = c_T = 5$.

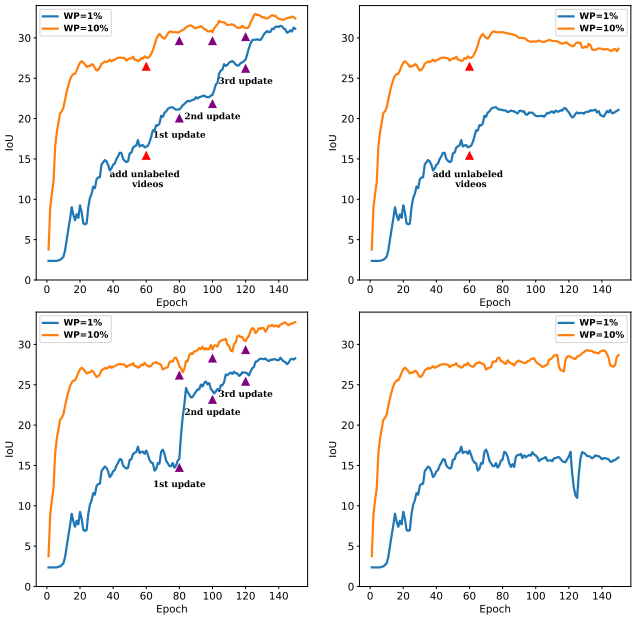


Figure 1. IoU of different methods on the Breakfast test set as a function of the number of training epochs. Top Left: SWSL+Self. Top Right: SWSL. Bottom Left: WSL+Self. Bottom Right: WSL.

## 4. Training Effects

In addition to the two methods shown in Figure 5 in the main paper, we include the performance of the two other methods (WSL+Self and WSL) on the Breakfast test set as a function of training epochs in Figure 1. If we do not use any unlabeled videos for training (WSL), the model will converge soon. However, if we generate pseudo-transcripts for unlabeled videos and add a subset of them to the weakly-labeled set (WSL+Self), the performance will be improved after each update, which demonstrates the effectiveness of self-training. Comparing WSL+Self with SWSL+Self, the improvement brought by using unlabeled videos of WSL+Self is not as significant as that of SWSL+Self when we have 1% of weakly-labeled videos. But the performance of the two methods is similar when we have 10% of weakly-labeled videos. This shows that our proposed approach can lead to more improvement when the number of weakly-labeled videos is small.

## 5. Limitation

Similar to most weakly-supervised action segmentation methods, our framework can only predict actions that occur in the weakly-labeled videos. We believe that our approach can be extended to address the open-set problem by adding an "unknown" class to accommodate new actions in test videos, which we leave as a future work.

## References


[1] J. B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[2] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[3] A. Miech, D. Zhukov, J. B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *International Conference on Computer Vision*, 2019. 2

[4] Y. Shen, L. Wang, and E. Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 2