Supplementary Materials: Set-Supervised Action Learning in Procedural Task Videos via Pairwise Order Consistency

Zijia Lu
Northeastern Univeristy
lu.zij@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

	Segmentation		Alignment	
	MoF	IoU	MoF	IoU
UM	25.4±15.1%	13.5±25.7%	26.1±11.7%	14.7±21.9%
SCT	26.2±17.9%	15.7±9.3%	29.1±18.3%	16.7±10.4%
POC	40.1±6.1%	32.5±6.9%	43.6±6.1%	35.8±5.4%

Table 1. Performance Statistics on the Breakfast dataset.

1. Additional Performance Statistics

As mentioned in the "Implementation Details" section of the paper, prior works on set-supervised action segmentation have only reported the best run results. However, the best run result can be unreliable for a non-robust model, which has large fluctuations in performance. In Table 1, we additionally show the "mean±Coefficient of Variation" for the results of POC and the replicated UM and SCT on the Breakfast dataset. Coefficient of variation equals to the standard deviation divided by mean, indicating the percentage of fluctuation in model performance. Notice that POC has a higher mean with *a lower coefficient of variation* in all cases, showing that a correct estimation of action ordering also benefits the model robustness.

2. Computational Complexity of POC

For training, the complexity of $\mathcal{L}_{\text{poc}}$ is $\mathcal{O}(T|\mathcal{A}|^2|\mathcal{V}|)$, to compute the ordering score $O(a_u, a_v)$ and ordering discrepancy $\pi(a_u, a_v)$ of each action pair in each video, where computing the scores for one pair only takes $\mathcal{O}(T)$. Here, T is the number of frames, $|\mathcal{A}|$ is the number of actions and $|\mathcal{V}|$ is the number of videos. Notice our complexity is linear in the length of videos, while that of three-step approaches (SCV/ACV) is $\mathcal{O}(T^2|\mathcal{A}|^2|\mathcal{V}|)$. On the other hand, the complexity is also linear in the number of videos, while ED/DTW conduct pairwise comparison between videos, leading to quadratic complexity in the number of videos. Although DTW allows using a barycenter, similar to our reference ordering, to reduce complexity, obtaining the barycenter needs solving additional iterative optimization [1, 2] with at least $\mathcal{O}(T^2|\mathcal{V}|)$ complexity per iteration. Moreover, even if the complexity of ED/DTW can be reduced, they still face the inherent issues discussed in the "Related Work" section of the paper.

For inference, the complexity of our method is $\mathcal{O}(T)$ to predict framewise labels while SCV and ACV have $\mathcal{O}(T^2|\mathcal{A}|K)$ complexity, where K is the number of hypothesized transcripts, often set to 1000. It takes 6 hours to train POC on one RTX 6000 GPU and 0.014 seconds to run inference on one video of Breakfast.

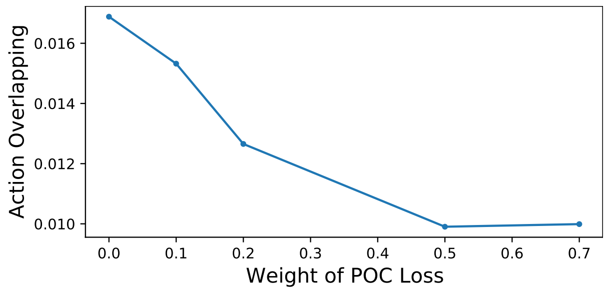

Figure 1. Overlap between actions, measured via the cosine similarity between their attentions, with respect to the weight of our POC loss. Results are computed on the first split of Breakfast.

3. Effect of POC Loss for Reducing Overlap among Actions

In this section, we show that the POC loss can not only enforce consistent ordering between actions, but also reduce their overlap, thus allowing our model to learn a distinct location for each action. First, we empirically show this point in Figure 1, where we plot the overlap between actions, measured by the cosine similarity of their attentions, with respect to the weight of the POC loss. Notice that increasing the loss weight successfully reduces the overlap. Next we provide the mathematical analysis of this claim.

In Equation (7) of the main paper, $\mathcal{L}_{\text{poc}}$ computes the average ordering discrepancy over all common action pairs.

For better readability, here, we consider a simplification of the POC loss, which only includes the terms related to one action pair (a_u, a_v) while excluding the irrelevant ones. We have $\mathcal{L}'_{\text{poc}} = \sum_{i\in\Lambda(a_u,a_v)} \pi^i(a_u, a_v)/|\Lambda(a_u, a_v)|$, where $\Lambda(a_u, a_v)$ is the set of videos containing both a_u and a_v, as defined in the main paper. The loss can be expanded as

$$\begin{aligned}
\mathcal{L}'_{\text{poc}} =& \frac{1}{|\Lambda(a_u,a_v)|} \sum_{i\in\Lambda(a_u,a_v)} \pi^i(a_u,a_v) \\
=& \frac{1}{|\Lambda(a_u,a_v)|} \sum_{i\in\Lambda(a_u,a_v)} [1 - O^\star(a_u,a_v)O^i(a_u,a_v) \\
& \qquad - O^\star(a_v,a_u)O^i(a_v,a_u)] \\
=& 1 - O^\star(a_u,a_v) \sum_{i\in\Lambda(a_u,a_v)} O^i(a_u,a_v)/|\Lambda(a_u,a_v)| \\
& \quad - O^\star(a_v,a_u) \sum_{i\in\Lambda(a_u,a_v)} O^i(a_v,a_u)/|\Lambda(a_u,a_v)| \\
=& 1 - O^\star(a_u,a_v)^2 - O^\star(a_v,a_u)^2 \\
=& 1 - [O^\star(a_u,a_v) + O^\star(a_v,a_u)]^2 \\
& \qquad + 2O^\star(a_u,a_v)O^\star(a_v,a_u). \qquad (1)
\end{aligned}$$

This loss has two terms, $O^\star(a_u, a_v)O^\star(a_v, a_u)$ and $-[O^\star(a_u, a_v) + O^\star(a_v, a_u)]^2$. First, minimizing $O^\star(a_u, a_v)O^\star(a_v, a_u)$ enforces ordering consistency, as we obtain the minimum when one of $O^\star(a_u, a_v), O^\star(a_v, a_u)$ is close to 1 and the other close to 0. Achieving this requires all videos having the same ordering of (a_u, a_v).

On the other hand, we show minimizing $-[O^\star(a_u, a_v) + O^\star(a_v, a_u)]^2$ reduces the overlap between actions. To do so, we first show that $O^i(a_u, a_v) + O^i(a_v, a_u) = 1 - \langle \boldsymbol{z}^i_{a_u}, \boldsymbol{z}^i_{a_v}\rangle$. Here $\boldsymbol{z}^i_{a_u,t} = \boldsymbol{W}^i_{a_u,t}/\tau^i_{a_u}$ can be viewed as the distribution of a_u's attention over temporal dimension in the video, with $\sum_t \boldsymbol{z}^i_{a_u,t} = 1$. We then have

$$\begin{aligned}
& O^i(a_u,a_v) + O^i(a_v,a_u) \\
=& \frac{1}{\tau^i_{a_u}} \sum_t \boldsymbol{W}^i_{a_u,t}\beta^i_{a_v,t} + \frac{1}{\tau^i_{a_v}} \sum_t \boldsymbol{W}^i_{a_v,t}\beta^i_{a_u,t} \\
=& \sum_t \boldsymbol{z}^i_{a_u,t} \frac{1}{\tau^i_{a_v}} \sum_{k=t+1} \boldsymbol{W}^i_{a_v,k} + \sum \boldsymbol{z}^i_{a_v,t} \frac{1}{\tau^i_{a_u}} \sum_{k=t+1} \boldsymbol{W}^i_{a_u,k} \\
=& \sum_t \boldsymbol{z}^i_{a_u,t} \sum_{k=t+1} \boldsymbol{z}^i_{a_v,k} + \sum \boldsymbol{z}^i_{a_v,t} \sum_{k=t+1} \boldsymbol{z}^i_{a_u,k} \\
=& \sum_t \boldsymbol{z}^i_{a_u,t} \sum_{k=t+1} \boldsymbol{z}^i_{a_v,k} + \sum \boldsymbol{z}^i_{a_u,t} \sum_{k=t-1} \boldsymbol{z}^i_{a_v,k} \\
=& \sum_t \boldsymbol{z}^i_{a_u,t}(\sum_{k=t+1} \boldsymbol{z}^i_{a_v,k} + \sum_{k=t-1} \boldsymbol{z}^i_{a_v,k}) \\
=& \sum_t \boldsymbol{z}^i_{a_u,t}(1 - \boldsymbol{z}^i_{a_v,t}) = 1 - \langle \boldsymbol{z}^i_{a_u}, \boldsymbol{z}^i_{a_v}\rangle. \qquad (2)
\end{aligned}$$

The value of $\langle \boldsymbol{z}^i_{a_u}, \boldsymbol{z}^i_{a_v}\rangle$ measures the correlation between the distributions of the attentions of a_u, a_v, thus reflects the overlap between them. With the observation, we can show

$$\begin{aligned}
& [O^\star(a_u,a_v) + O^\star(a_v,a_u)]^2 \\
=& \left(\sum_{i\in\Lambda(a_u,a_v)} \frac{O^i(a_u,a_v)}{|\Lambda(a_u,a_v)|} + \sum_{i\in\Lambda(a_v,a_u)} \frac{O^i(a_v,a_u)}{|\Lambda(a_u,a_v)|} \right)^2 \\
=& \left(\sum_{i\in\Lambda(a_u,a_v)} (O^i(a_u,a_v) + O^i(a_v,a_u))/|\Lambda(a_u,a_v)| \right)^2 \\
=& \left(1 - \sum_{i\in\Lambda(a_u,a_v)} \langle \boldsymbol{z}^i_{a_u}, \boldsymbol{z}^i_{a_v}\rangle/|\Lambda(a_u,a_v)| \right)^2. \qquad (3)
\end{aligned}$$

Therefore, minimizing $-[O^\star(a_u, a_v) + O^\star(a_v, a_u)]^2$ will reduce $\langle \boldsymbol{z}^i_{a_u}, \boldsymbol{z}^i_{a_v}\rangle$ for each video thus reduces the overlap between the actions.

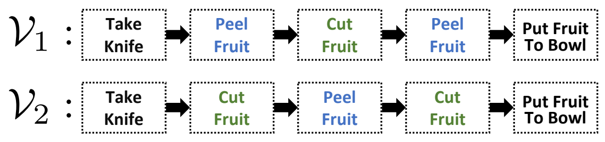

Figure 2. Two videos with repeated actions from the *salad* recipe in Breakfast.

4. POC Loss on Repeated Actions

Repeated actions that occur multiple times in a video impose a difficulty in learning action ordering, as the ordering between them and other actions is often ambiguous. Yet we show our ordering score defined in Equation (2) of the main paper can also handle repeated actions.

To illustrate this, in Figure 2, we show transcripts of two videos with repeated actions. Notice that in both videos, there is not a single ordering for *peel fruit*, *cut fruit*) as each action could occur before or after the other in each video. One workaround is to separately consider the ordering of each occurrence. Yet, it is difficult to find an alignment for different occurrences of an action across videos. In contrast, computing our ordering score on two videos gives O^1(*peel fruit*, *cut fruit*) $= O^2$(*peel fruit*, *cut fruit*) $= 0.5$. As a result, the reference ordering between the actions is also 0.5, meaning that we consider both orderings as plausible. Therefore, our method allows to handle ambiguous ordering of the repeated actions rather than artificially finding an ordering, while trying to estimate an unambiguous ordering, e.g., the ordering of (*take knife, cut fruit*), still helps in action localization.

5. POC Loss on Varied Action Orderings

Another challenge of learning action ordering comes from the action pairs with varied ordering across videos.

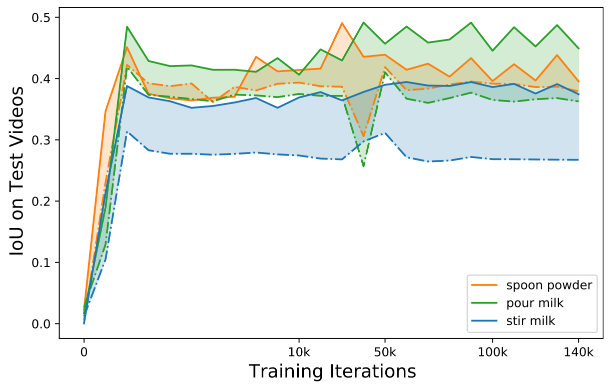

Figure 3. Test performance in terms of training iterations on three actions from *milk* recipe of Breakfast dataset. The solid and dashed lines show the scores for our models with and without POC loss, respectively, and the shaded area shows the gap between their performance.

For example, two actions a_u, a_v may occur as a_u before a_v in $\mathcal{V}_1$ and a_v before a_u in $\mathcal{V}_2$. Our $\mathcal{L}_{\text{poc}}$ can also properly handle the two actions, as in this case, the reference ordering $O^\star(a_u, a_v), O^\star(a_v, a_u)$ will be 0.5, meaning it is uncertain about the ordering, hence allowing both as desired, while the ordering between a_u/a_v and other actions still helps localization.

In Figure 3, we show the IoU on test videos as a function of training iterations for three actions of the *making chocolate milk* recipe on Breakfast. The solid and dashed lines show the scores for our models with and without POC loss, respectively, and the shaded area shows the gap between their performance. It can be observed that, for *stir milk* that has a consistent ordering with other actions, its IoU is improved by 12% with POC loss. More importantly, for $\{$*spoon powder*, *pour milk*$\}$ with different orderings, where people *spoon powder* before *pour milk* in 64% of videos and perform the opposite in the remaining 36% of the videos, POC loss still boosts the IoU by 5% and 10%, respectively, showing its capacity to handle varied action orderings.

6. POC for Transcript-Supervised Learning

As mentioned in Section 4.1 of the main paper, our method can address transcript-supervised action learning. To do so, we modify the POC loss to compute a reference ordering for each video from its ground-truth transcript. Specifically, let $\Upsilon^i = \{a_1, a_2, a_3, ...\}$ denote the transcript of a video. We first transform it into a one-hot label matrix $\boldsymbol{W}^{\star,i} \in \boldsymbol{R}^{A\times|\Upsilon^i|}$ with $\boldsymbol{W}^{\star,i}_{a_j,j} = 1$ and 0 elsewhere. Based on it, we can compute the true reference ordering, $O^{\star,i}$, of this video according to Section 3.2.2 of the paper. We use $O^{\star,i}$ to compute the ordering discrepancy and the POC loss via Equations (6) and (7) in the paper, respectively.

	Segmentation		Alignment		Attention	Action
	MoF	IoU	MoF	IoU	entropy	overlap
Eq. (5)	34.2	30.1	35.9	31.4	0.48	0.09
Eq. (4)	40.1	32.5	43.6	35.8	0.20	0.02

Table 2. Average model performance over runs for different computation of video-level action feature on all splits of Breakfast. Action overlap is measured as the average cosine similarity between the attentions of actions.

7. Effect of Video-Level Features

In this section, we compare the effect of different methods to compute the video-level feature of an action. As mentioned in Section 3.2.3 of the main paper, to recognize if an action is present in a video with $\mathcal{F}^{\text{cls}}$, we compute a video-level feature for an action a

$$\boldsymbol{g}_a = \frac{1}{T'}\sum_t \boldsymbol{W}_{a,t}\boldsymbol{h}_t, \tag{4}$$

while some prior works [3,4] compute the feature as

$$\boldsymbol{g}'_a = \frac{\sum_t \boldsymbol{W}_{a,t}\boldsymbol{h}_t}{\sum_t \boldsymbol{W}_{a,t}}. \tag{5}$$

In Table 2, we show that using (4) significantly improves the results over (5). This comes from the fact that while both methods guide the attention of an action to concentrate on the correct region, (5) fails to ensure the magnitude of the action's attention in the region is larger than those of other actions, which is vital for accurate framewise predictions. Specifically, in (5), the overall magnitude of $\boldsymbol{W}_a$ does not affect $\boldsymbol{g}'_a$. It means that $\boldsymbol{W}_a$ close to zero that focuses on the correct region can lead to accurate recognition of the action. However, action segmentation based on it will yield very low accuracy. In contrast, (4) allows a small $\boldsymbol{W}_a$ to be penalized by the ranking loss $\mathcal{L}_{\text{v-rk}}$, because such a $\boldsymbol{W}_a$ results in a close-to-zero norm of $\boldsymbol{g}_a$ and a small value of $\mathcal{F}^{\text{cls}}(\boldsymbol{g}_a)$, since $\mathcal{F}^{\text{cls}}$ being a multi-layer perceptron is generally linear w.r.t. the input. $\mathcal{L}_{\text{v-rk}}$ will increase $\mathcal{F}^{\text{cls}}(\boldsymbol{g}_a)$ for positive actions, thus increases $\boldsymbol{W}_a$ as well, which in turn improves action segmentation. Moreover, increasing $\boldsymbol{W}_a$ also has the effect of decreasing the entropy of attentions on a frame, thus reducing action overlaps, which can be seen in the last two columns of Table 2.

References

[1] M. Cuturi and M. Blondel. Soft-dtw: a differentiable loss function for time-series. *International Conference on Machine learning*, 2017.

[2] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 2011.

[3] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. *CVPR*, 2020.

[4] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, and Junsong Yuan. Two-stream consensus network for weakly-supervised temporal action localization. *ECCV*, 2020.