

Pose-Aware Weakly-Supervised Action Segmentation

Supplementary Material

1. More Implementation Details

1.1. Implementation Details on Pose Normalization

Given a frame at time t , raw pose $p_t \in \mathbb{Z}^{K \times 2}$ is a collection of (x, y) coordinates for K human keypoints. Here, K represents the number of 2D keypoints extracted by an external pose extractor and \mathbb{Z} is the set of integers. Before inputting these raw keypoints to the pose encoder, we perform a normalization step to ensure they are unaffected by changes in perspective, rotation, and positional offset in the frame. Specifically, each keypoint is centered and scaled with respect to the "center of mass" of the human, which is determined by averaging the coordinates of all joints. Subsequently, we determine the angle required to rotate each adjusted keypoint so that the head and "center of mass" align vertically, sharing the same x coordinates. The specific mathematical formulation is listed below:

$$\begin{aligned} \text{centroid} &= \frac{1}{K} \sum_{i=1}^K p_{t_i} \\ \text{centered_pose} &= p_t - \text{centroid} \\ \text{avg_distance} &= \frac{1}{K} \sum_{i=1}^K \sqrt{(x_{t_i} - x_{\text{centroid}})^2 + (y_{t_i} - y_{\text{centroid}})^2} \\ \text{scaled_pose} &= \frac{\text{centered_pose}}{\text{avg_distance}} \\ \text{angle} &= \arctan \left(\frac{y_{\text{scaled_pose_of_head_joint}}}{x_{\text{scaled_pose_of_head_joint}}} \right) \\ \text{rotation_matrix} &= \begin{bmatrix} \cos(\text{angle}) & -\sin(\text{angle}) \\ \sin(\text{angle}) & \cos(\text{angle}) \end{bmatrix} \\ \text{normalized_pose} &= \text{scaled_pose} \times \text{rotation_matrix} \end{aligned}$$

These normalized 2D keypoints, \bar{p}_t , are then fed into the pose encoder.

Our pose network is a fully-connected MLP network with sizes [34, 128, 128, output_size], where the output_size is determined by the specific network architecture we use in training.

1.2. Implementation Details on different backbones

As mentioned in the main paper, we extracted I3D [1] features from ATA and IKEA datasets, and for Desktop Assembly, we used ResNet [4] features. The dimension for I3D features in ATA dataset is 2048, whereas in IKEA is 400. The dimension of ResNet feature in Desktop dataset is 512.

In experiments with DP[3] as the baseline, we modeled the video encoder with Transformers. The projection network for video feature is a fully-connected layer of input size that is determined by the input dimension of video features and output size of 128. We set the pose network to have input size of 34 and output size of 128 to have a matched dimension for contrastive learning. During inference time, the projection and pose networks are not used. The detailed parameters of network structure are not changed. In our experimentation, the learning rate is set to 0.01, beam size is 151, window size is 15. During evaluation, we use the default exploration threshold of 0.7 for our segmentation results on ATA dataset. Also, we set an exploration threshold of 0.0 for IKEA and Desktop datasets due to their similar training and test transcripts. The training iteration is 40000 for ATA dataset, 20000 for IKEA dataset, and 10000 for Desktop dataset.

In experiments with TASL[6], we regard the existing GRU network as the output for RGB embedding. The output dimension of RGB embedding is 64, so we set the pose network to have input size of 34 and output size of 64 to perform contrastive learning. In our experimentation, we simply add the contrastive learning loss without any network modification. Specifically, in the TASL architecture, the learning rate is 0.01, decode sample rate is 30, window size is 33, space size is 10, pred size is 3, auto encoder weight is 0.2, edge window is 6 and edge step is set to 2. The training iteration is 20000 for ATA dataset and 6000 for Desktop Dataset.

For MuCon[7], we pass the scaled pose keypoints to the pose encoder to obtain pose embeddings of size 2048, corresponding to the RGB embeddings. These RGB embeddings are produced by a multi-stage temporal convolutional network [2]. However, we pass the pose embeddings to a "frozen" copy of the temporal convolutional network to obtain pose embeddings that correspond to the same format as the RGB embeddings, i.e., same number of embeddings in time and same dimensionality. Then, we perform the contrastive learning on these embeddings for both pose and RGB modalities. In our experiments, we train for 100 epochs for both the baseline and our method. The specific parameters are set to their default values with learning rate of 0.01, and momentum of 0.0. It is noteworthy to mention that MuCon has three output versions, and we picked the best version (MuCon-full) for our comparisons.

Table 1. Split-wise comparison of proposed method versus baseline on IKEA dataset for online action segmentation.

Metric	acc	IoU	Edit	F1@0.5
Split	1/2/3/4/5	1/2/3/4/5	1/2/3/4/5	1/2/3/4/5
Greedy [5]	54.4/60.1/ 50.9 /54.9/45.1	28.5/30.8/26.2/ 29.6 /20.2	48.3 /46.7/37.4/42.2/33.0	22.7/28.1/21.8/ 26.1 /19.7
DP [3]	56.6/59.6/50.2/51.8/ 53.1	28.3/30.7/ 26.3 /26.2/ 24.9	46.8/ 55.3 /46.2/47.2/ 45.0	24.9/29.9/ 24.5 /25.0/ 25.9
DP + Ours	57.3 / 61.7 /50.3/51.3/51.4	29.9 / 31.5 / 26.3 /26.6/24.2	48.3 / 55.3 / 46.3 / 47.6 /44.6	25.9 / 30.2 /24.2/25.2/25.5

2. Experimental Results on IKEA Dataset

As mentioned in the main paper, we provide split-wise results in Table 1. The overall results in the main paper are computed as the average of all splits. We associate the overall marginal improvements on the IKEA dataset mostly to its 5th split. For other splits, single-person is mostly exhibited in the training and testing sets. On the contrary, in many videos of the 5th split, the single-person assumption is violated by background people, which negatively impacts our pose encoding accuracy. While our contrastive learning module only establishes RGB-pose correspondence for each person, the pose encoding might not be so accurate when there are multiple persons in background. Results of split three and split four are competitive between our method and the baseline, whereas splits one and two exhibit the largest improvements of our proposed pose-infused methodology. In general, our method beats previous baselines in most cases in the IKEA dataset over different metrics and splits.

References

- [1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017. 1
- [2] Yazan Abu Farha and Jurgen Gall. Ms-ten: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019. 1
- [3] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, and Behzad Dariush. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10128–10138, 2023. 1, 2
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1
- [5] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6243–6251, 2019. 2
- [6] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *Proceedings of the IEEE/CVF Inter-*

national Conference on Computer Vision, pages 8085–8095, 2021. 1

- [7] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juerge Gall. Fast weakly supervised action segmentation using mutual consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1