Supplementary materials for OTAS

Anonymous WACV Algorithms Track submission

Paper ID 62

1. Implementation Details

1.1. Global Perception Module

Architecture. The spatial encoder is a ResNet50 model [9] with 2048 output classes. The temporal decoder is adapted from the Transformer [23] with 2048 hidden dimensions and 8 heads. The decoder is constructed by stacking 6 up-sampling blocks. Each block contains an up-sampling function, a convolution layer, and a ReLU activation function. The kernel size is 7 for the first and last blocks and 3 for the rest. The interpolation sizes are 8, 16, 32, 64, 128, and 256, sequentially. The channels are 1024, 512, 256, 128, 64, and 3, respectively.

Optimizer and schedule. We use the standard Adam [10] optimizer with a learning rate of 1e-4 and a multi-step scheduler. We train the model for a total of 250000 steps with a batch size of 16.

1.2. Human-Object Interaction Model

Architecture. The architecture of the human-object interaction model is the same as the global perception model without the decoder.

Human-object interaction masks. We obtain the masks from an off-the-shelf object detection model implemented by the open source platform Detectron2 [28]. We select the Faster-RCNN-X101-FPN model pre-trained on the COCO train2017 dataset [17] with a box average precision (box AP) of 43.0. For human-object interaction, we select masks that contain human body parts, *e.g. person*.

1.3. Object Relationship Model

Architecture. The encoder is also a ResNet50 model [9], and the decoder is the same as the frame prediction model. We adapt architecture from GATv2 [3] for graph implementation in the bottleneck part. We use two 8-heads self-attention layers, with 32 input channels and 6 output channels for each head. We then add a fully-connected layer to project the output to 2048 dimension.

Optimizer and schedule. We use the standard Adam [10] optimizer with a learning rate of 5e-5 and a multi-step scheduler. We train the model for a total of 100000 steps with a batch size of 16.

Object relationship look-up table. For all the object classes in COCO dataset [17], we select 44 classes that appear most frequently in instructional videos. All 44 classes are depicted in Table 1. We then seek their relations through human annotations from the Visual Genome dataset [11], which is designed for cognitive tasks. To be more specific, we first re-organize the object classes of the Visual Genome dataset to be in line with the 44 classes we selected from the COCO dataset. Some examples of the re-organization are shown in Table 2. Then, for each class, we list and count all possible connections of the objects through predicates provided by the Visual Genome dataset. Finally, we filter out object pairs that appear less than 30 times and build the object relation look-up table. A few illustrations of the look-up table are illustrated in Table 3.

toothbrush	scissors	vase	clock	book	refrigerator	sink
toaster	oven	microwave	cell	keyboard	remote	mouse
laptop	tv	table	plant	couch	chair	cake
donut	pizza	hotdog	carrot	broccoli	orange	sandwich
apple	banana	bowl	spoon	knife	fork	cup
glass	bottle	suitcase	handbag	backpack	bench	person
rack	cabinet					

Table 1. **New object classes selected from the COCO dataset.**

Object masks. We obtain the object masks from the same off-the-shelf object detection model [28] as the human-object interaction model. We select masks that are in the new object classes and have confidence scores that are larger than 0.7.

person
person, man, woman
table
table, coffee_table, counter, countertop, desk
knife
knife, steak_knife, butter_knife, knife_blade, bread_knife, butcher_knife

Table 2. **Illustrations of the re-organization for the Visual Genome dataset.**

toothbrush
cup, sink, person, rack, table
cake
table, bowl, person, cup, knife, fork
knife
table, fork, person, cake, pizza, apple, orange, banana, sandwich

Table 3. **Illustrations of the object relationship look-up table.**

2. Evaluation Metrics

2.1. F1 Score

For the computation of the F1 score, we follow the implementation of Shou et al. [20], Wang et al. [25] and first calculate the distance between the N detected boundaries and the M ground truth boundaries. We pair each ground truth boundary with a detected boundary that has a minimal distance. Then, we set a fixed distance threshold to determine if the detected boundary is positive or not. The total number of positive detection is P. The Precision/Recall and F1 score can be computed as:

$$\begin{aligned} \text{precision} &= \frac{P}{N} \\ \text{recall} &= \frac{P}{M} \\ F_1 &= 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \end{aligned}$$

We compute the Precision/Recall and F1 score for each video and average across the whole dataset.

As mentioned in the paper, previous works [20, 25] set the distance threshold to be 5% of the length of the corresponding video instance, while we choose 2 seconds which is invariant of video lengths. We show some examples in Figure 1 for the impact of the 2 different thresholds. It is clear that the small threshold is more suitable and general for the evaluation of various instructional videos.

2.2. Hungarian Matching

To perform a fair evaluation with previous methods utilizing clustering algorithms [4, 6, 14, 18], we first applying clustering algorithm as in Du et al. [6] to transfer OTAS boundaries into clusters based on IDT features. Then, we follow [1, 25] and perform the Hungarian matching [13] on a video level.

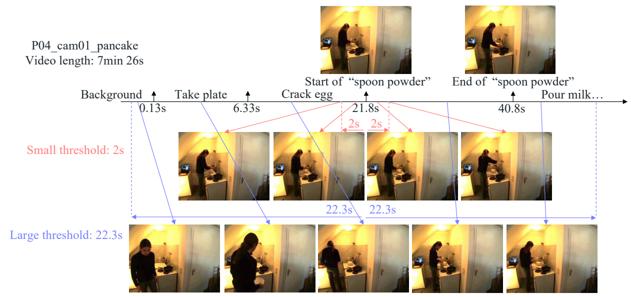

Figure 1. **Illustrations of different thresholds for the boundary-level F1 score.** For evaluation of the boundary "Start of spoon powder", it is clear that a 2s deviation is not harmful. However, a 22.3s threshold (5% of a 7 min 26s video) will cause the boundaries even in "Background" and "Take plate" to be falsely labeled positive.

Noting that for other clustering-based methods that are either only performing on same activities [7, 8, 14, 16, 19, 22, 24, 26, 27] or extend to unknown activities but only provide global-level Hungarian matching results and do not provide code to reproduce [5, 15], we can not conduct a fair comparison.

2.3. Mean over Frames (MoF)

We calculate MoF after clustering and Hungarian Matching. MoF indicates the percentage of frames in the video instance that are correctly segmented [14, 19]. For a video with K frames, we count all the correct frames C and compute the MoF as:

$$\text{MoF} = \frac{C}{K}$$

We average the video-wise MoF across the whole dataset.

3. User Study

3.1. Implementation

We first pick 20 videos from the Breakfast dataset [12] randomly and generate segmented videos from 5 different methods: one from ground truth, one from OTAS, one from ABD [6], one from CTE [14], and the last one from TW-FINCH [18]. For each video, We shuffle and label 5 segmentation results with numbers 1-5. We invite 33 users to watch and rank the segmentation results with only the reference to the original videos. A part of the user study questionnaire is depicted in Figure 2. Since it is a temporal segmentation task, the average time of completion is 2.5 hours. We use $6 - rank$ as the score for each method (i.e., rank No.1 has 5 points). A video-wise score distribution is shown in Figure 8. We show one of our segmentation results that gains the highest score in Figure 4.

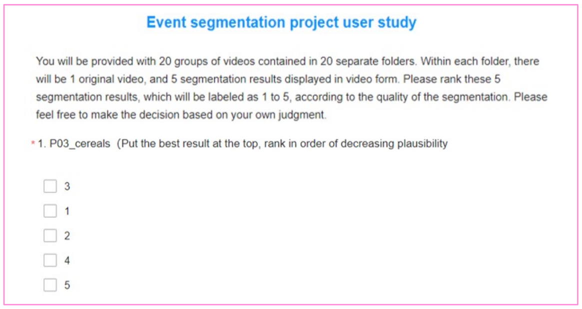

Figure 2. **User study questionnaire interface.** We provide only the options to choose from, excluding any reference to the granularity information.

3.2. Breakfast Ground Truth

We provide more illustrations of the inconsistent ground truth segmentation of the Breakfast dataset [12] in Figure 3.

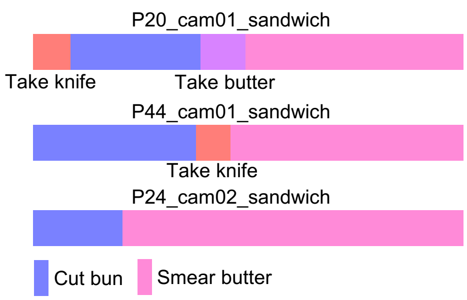

Figure 3. **Inconsistency of the ground-truth.** For "Sandwich" activity, the action "Cut bun" can be further segment into "Take bun", "Take knife", and "Actually cut the bun"; while the action "Smear butter" can be further segment into "Take butter", "Take knife", and "Actually smear the butter". However, the ground truth annotation provides inconsistent segmentation that sometimes produces larger segments and sometimes smaller segments. Even within a video, the segmentation is inconsistent.

4. Qualitative Result

For a better illustration of boundary evaluation, we assign all different ground truth segments distinct colors within a video regardless of labels.

4.1. Breakfast

We provide more qualitative comparisons with the ground truth of our methods on the Breakfast dataset [12] in Figure 5.

4.2. 50Salads

The qualitative comparison of our methods on both eval-level and mid-level 50salads [21] is illustrated in Figure 6. For eval-level, we compare with baselines ABD [6], CTE [14], TW-FINCH [18], Coseg [25] and groundtruth. For mid-level, we only compare with ABD [6], CTE [14], TW-FINCH [18], and ground truth, since Coseg [25] does not provide mid-level results.

4.3. INRIA

The INRIA dataset [2] is collected from YouTube and segmented with the aid of English transcripts obtained from YouTube's automatic speech recognition (ASR) system. For all tasks, the ordered sequence of ground truth steps is made by an agreement of 2-3 annotators who have watched the input videos and verified the steps on instruction video websites. Therefore, the rest of the video where no step is assigned is considered background. The percentage of average background frames is 73% of all frames. The background frames are various and complicated, as shown in Figure 7. Since we rely on feature differences for boundary detection, the variation in backgrounds influences the result largely. Moreover, we do not have access to prior knowledge of cluster numbers. Therefore, the result of INRIA is very likely to be over-segmented.

5. Ablation Study

5.1. Comparison of Different α

We utilize a hyper-parameter α to control the number of boundaries. The comparison of different α is shown in Tab. 4. Generally, lower α leads to higher recall, but also redundancy, which causes precision to drop. Higher α generates fewer boundaries, resulting in higher precision and MoF but low recall. We select $\alpha = 15$ that best balances the trade-off.

	F1(*small*)	Recall(small)	Precision(small)	MoF
$\alpha = 8$	42.43	**71.96**	30.08	65.22
$\alpha = 25$	42.77	48.31	**38.37**	67.57
$\alpha = 15$	**44.49**	53.90	37.87	**67.90**

Table 4. **Comparison of different αs.** There is a trade-off between better precision and better recall.

5.2. Global Perception Module Architectures

We also conduct an ablation study on different model architectures for the global perception module. Specifically, we leverage features from pre-computed IDT, a pre-trained ResNet-50 model and ResNet with a 2-layer LSTM model that respectively replaces the Transformer layer for comparison. The results demonstrated in Table 5 indicate that the Transformer-based model generates finer features for action segmentation than the other models.

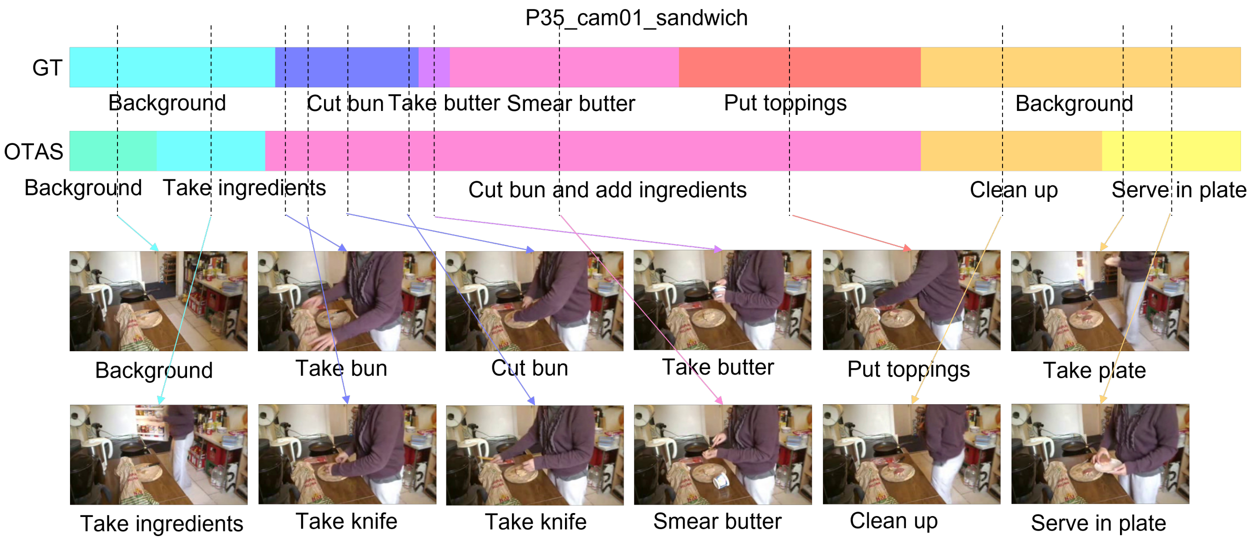

Figure 4. **More consistent granularity of the segmentation results produced by OTAS.** The video shown in the figure contains several smaller segments at the action level. The ground truth only segments "Take butter" out, and combine the others, which is confusing while watching. Furthermore, the ground truth does not separate "Take ingredients" and "Serve on plate" from the backgrounds. However, our segmentation result is neat and consistent, which is more in line with human consensus.

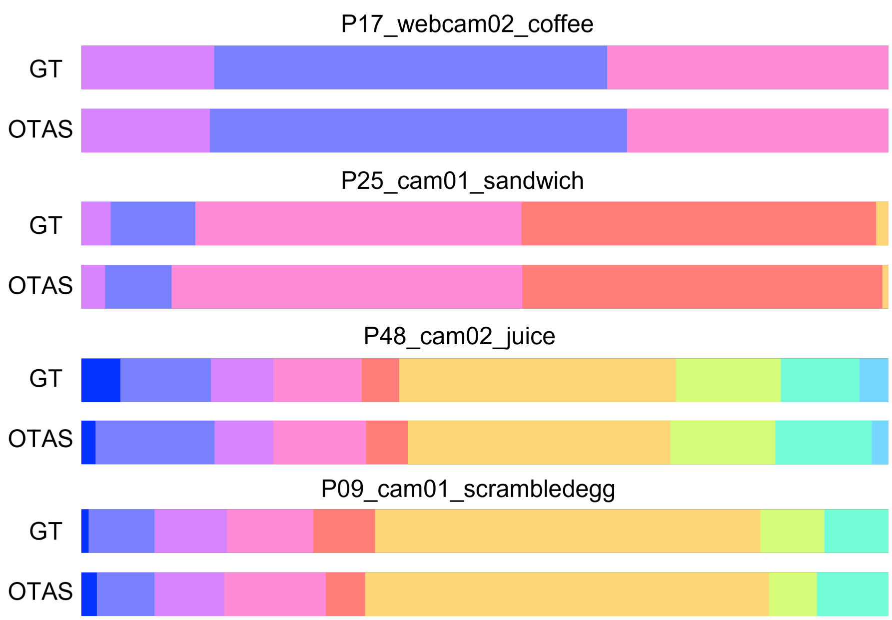

Figure 5. **Qualitative comparison with ground truth (GT) of the Breakfast dataset.** The results predicted by OTAS are largely in line with the ground truth.

	F1(*small*)	MoF
IDT	27.49	63.50
Pretrained-ResNet	35.28	65.00
LSTM	36.00	65.50
Transformer	**37.46**	**65.99**

Table 5. **Ablation of different architectures for the global perception module (OTAS excluding the local attention module) on Breakfast.** Transformer-based approach achieves the best performance of F1 score and IoU.

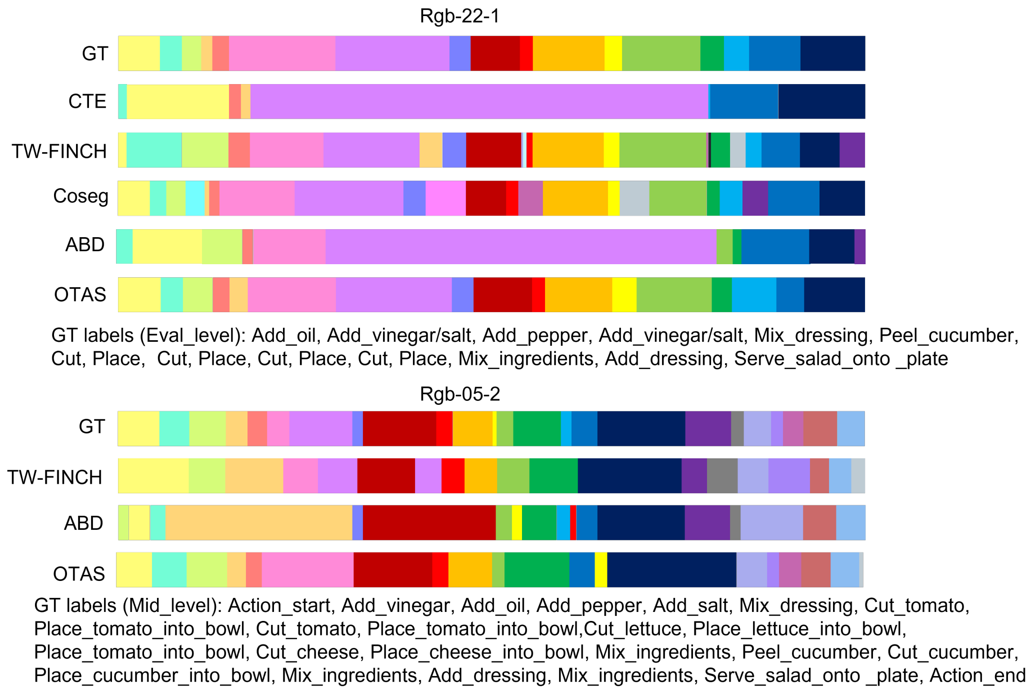

Figure 6. **Qualitative comparison of the 50Salads dataset.** Note that the original illustration of Coseg is not aligned with the actual timestamp. However, since they do not provide code to reproduce, we roughly resize their illustration for comparison.

Figure 7. **Illustration of the various background frames of INRIA.** It contains frames when the person shows preparation, stops to introduce upcoming steps, illustrates precautions, etc. It also contains shot changes and video editing.

WACV #62

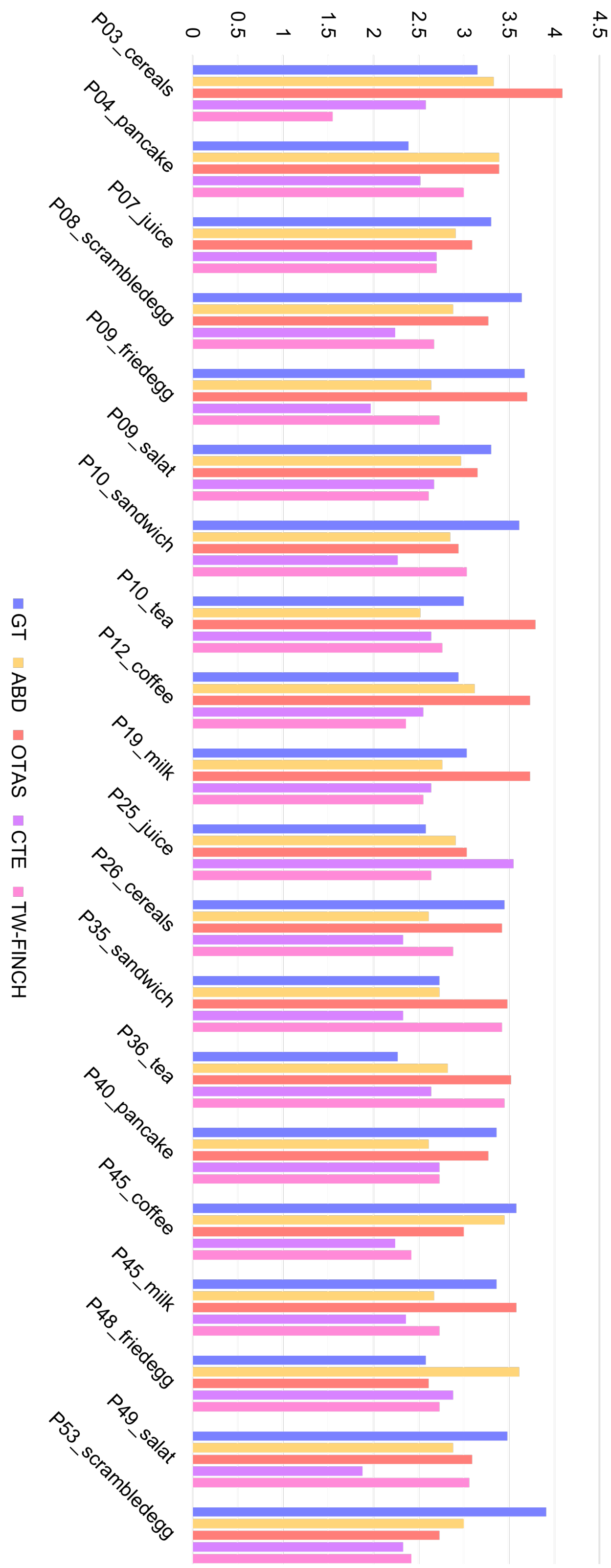

Figure 8. **Video-wise user study score.**

References

[1] Sathyanarayanan N. Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 3

[3] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 1

[4] Guilherme de AP Marques, Antonio José G Busson, Álan Lívio V Guedes, and Sérgio Colcher. A cluster-based method for action segmentation using spatio-temporal and positional encoded embeddings. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, pages 181–187, 2021. 2

[5] Guodong Ding and Angela Yao. Temporal action segmentation with high-level complex activity labels. *arXiv preprint arXiv:2108.06706*, 2021. 2

[6] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3323–3332, 2022. 2, 3

[7] Ehsan Elhamifar and Dat Huynh. Self-supervised multi-task procedure learning from instructional videos. In *European Conference on Computer Vision*, pages 557–573. Springer, 2020. 2

[8] Ehsan Elhamifar and Zwe Naing. Unsupervised procedure learning via joint dynamic summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6341–6350, 2019. 2

[9] 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*, pages 770–778, 2016. 1

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[11] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 1

[12] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 2, 3

[13] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[14] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12066–12074, 2019. 2, 3

[15] Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised activity segmentation by joint representation learning and online clustering. *arXiv preprint arXiv:2105.13353*, 2021. 2

[16] Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12628–12636, 2021. 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[18] Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11225–11234, 2021. 2, 3

[19] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8368–8376, 2018. 2

[20] Mike Zheng Shou, Stan Weixian Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8075–8084, 2021. 2

[21] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738, 2013. 3

[22] Sirnam Swetha, Hilde Kuehne, Yogesh S Rawat, and Mubarak Shah. Unsupervised discriminative embedding for sub-action learning in complex activities. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2588–2592. IEEE, 2021. 2

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[24] Rosaura G VidalMata, Walter J Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1238–1247, 2021. 2

[25] Xiao Wang, Jingen Liu, Tao Mei, and Jiebo Luo. Coseg: Cognitively inspired unsupervised generic event segmentation. *arXiv preprint arXiv:2109.15170*, 2021. 2, 3

[26] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Unsupervised action segmentation with self-supervised feature learning and co-occurrence parsing. *arXiv e-prints*, pages arXiv–

2105, 2021. 2

[27] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1819–1828, 2022. 2

[28] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 1