# Supplemental Material ActionVOS: Actions as Prompts for Video Object Segmentation

Liangyang Ouyang[1], Ruicong Liu[1], Yifei Huang[1]*, Ryosuke Furuta[1], and Yoichi Sato[1]

The University of Tokyo
{oyly, lruicong, hyf, furuta, ysato}@iis.u-tokyo.ac.jp

## 1 Dataset Details

**VISOR [3]** is a new dataset conducted on EPIC-KITCHENS [1,2] suitable for segmenting hands and active objects in egocentric videos. We use their videos and annotations for both training and validation. We exclude videos annotated with less than 2 frames, resulting in a total of 3,238 videos after pre-processing. In the validation set, we randomly choose 330 action clips and manually annotate the positive and negative objects. Tabs. 1,3,4,5,6 are based on this validation set.
**VOST [7]** is a recent dataset collected for video object segmentation under transformations. We only use VOST for validation since only one object class is annotated for each video. VOST annotate multiple instances and we treat all instances within the same video as one active object.
**VSCOS [9]** is constructed recently by selecting state-changing videos from EPIC-KITCHENS [1,2]. We also only use VSCOS for validation of state-changed objects. As it shares multiple video clips with VISOR, we filter out the video clips that have appeared in the training set of VISOR to avoid data leakage.

For all three datasets, we adhere to their original split rules for dividing the train-valid set. After the pre-processing, we obtain 13,205 videos and 76,873 objects for training, 467 videos and 1,841 objects for validation. The validation sets contain 1,133 positive and 708 negative objects.
**Annotation Tool for VISOR.** As shown in Fig. 10, we developed a Django [4] application as the annotation tool. On a single interface, we provide two RGB frames with object masks and one action narration. Then, annotators are asked to keep only positive object masks by checking and un-checking the check-boxes associated with object names. Fig. 10 (b) illustrates an annotation example, where only positive objects are checked, *e.g.*, "tofu", "tofu container". Before beginning the annotation process, annotators would read the annotation rules along with 15 examples. After all annotators finished their tasks, cases labeled as "Missing Objects" and "Other Errors" are excluded, while interacted instances in "Redundant Instances" cases are manually retained.
**Annotation Rules.** As discussed in Sec. 3, we define active objects, hands, hand-tools, containers, surfaces and contents as the positive objects in Action-

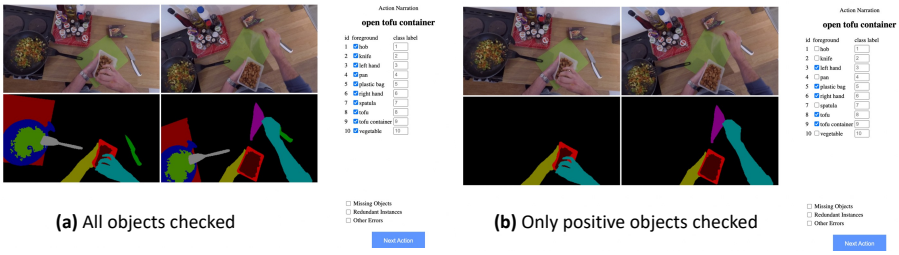


**Fig. 10:** The annotation tool interface.

VOS. In manual annotation, we extend the definition to detailed rules. We show every annotator the annotation rules below:

1) Objects described by the action narration are all positive. We name them narration positive objects.
2) Hands/gloves performing the action are positive. Hand not used for this action is negative. If you could not decide which hand is used, keep both hands positive.
3) Hand-tools used in the action are positive. If you could not decide if the tool is a hand-tool, keep it positive.
4) The lid/cover/container/surface moving with narration positive objects by the action is positive.
5) If a container/surface is a narration positive object, the contents moving with the container/surface by the action is positive. The contents not moved with the container/surface by the action is negative.
6) If the narration positive objects are all missing in the images, choose "Missing Objects" checkbox.
7) If there are redundant instances in the same class but not interacted in the action, choose "Redundant Instances" checkbox.
8) If there are other reasons for failing the annotation, choose "Other Errors" checkbox.

## 2 Evaluation Metrics

**mIoU and cIoU.** mIoU and cIoU are widely-used in segmentation tasks [5, 6]. mIoU calculates mean intersection over union while cIoU calculates total intersection pixels over total union pixels. As ActionVOS introduces a novel concept of distinguishing positive and negative objects, we report IoUs separately for positive and negative objects, *i.e.*, p-mIoU, n-mIoU, p-cIoU, n-cIoU.
**gIoU.** gIoU is introduced in [6] to combine the segmentation result and a no-target classification result. In our work, this metric simultaneously evaluates the ability to segment positive objects and distinguish negative objects.
**Acc.** We further use classification accuracy to evaluate the model's performance in identifying active objects. It is calculated by binary classification results, Acc$=\frac{TN+TP}{TN+TP+FN+FP}$.

## 3 Action Vocabulary

**Vocabulary statistics** of training [3] and validation [3,7,9] sets with the number of unseen categories are provided in Tab. 7.

**Table 7:** Vocabulary statistics.

| Split | # actions | # verbs | # nouns |
|---|---|---|---|
| Training [3] | 1898 | 90 | 242 |
| Validation [3,7,9] | 187 / 39 / 37 | 43 / 20 / 9 | 125 / 32 / 33 |
| Unseen in validation | 37 / 26 / 9 | 0 / 3 / 0 | 4 / 16 / 2 |

**Evaluation on unseen categories.** We compare ActionVOS model performance with other baselines on unseen actions in Tab. 8, where our method achieved best results. This is because the ActionVOS model not only identifies target objects through input object names, but also learn to segment active objects through human action interactions. For example, in the last visualization in Fig. 6, neither "paint" nor "nail" appear in the training set, while ActionVOS with action prompts still successfully segmented the painted nail.

**Table 8:** Evaluation on unseen actions.

| Method | p-mIoU ↑ | n-mIoU ↓ | p-cIoU ↑ | n-cIoU ↓ | gIoU ↑ | Acc ↑ | VOST p-mIoU | VOST p-cIoU | VSCOS p-mIoU | VSCOS p-cIoU |
|---|---|---|---|---|---|---|---|---|---|---|
| RVOS [8] | 60.0 | 49.0 | 63.5 | 63.6 | 42.9 | 65.3 | 18.6 | 12.6 | 31.5 | 21.4 |
| HOS [3] | 51.9 | **9.0** | 57.3 | **6.4** | 64.9 | 72.0 | 13.6 | 11.4 | 42.7 | 38.8 |
| ActionVOS | **60.3** | 21.0 | **65.7** | 39.7 | **66.1** | **79.7** | **22.5** | **18.0** | **44.9** | **43.1** |

**Hard action categories**. The actions with segmentation p-mIoU lower than 30% in VISOR validation set are listed below: "put down pakage", "dry hand", "put tea towel", "push oven tray", "pour-into water", "sprinkle-on salt", "take-out grape", "take carrot bag", "pick-up spinach", "get meat mix". In VOST validation set, "cut paper" and "divide dough" get lowest p-mIoU. We find that **invisible hands, ambiguous object names and significant shape change** bring low ActionVOS results. The visualization of typical fail cases are shown in Fig. 11.


**Fig. 11:** Visualization of ActionVOS failed cases.

## 4 Video Visualization

Please refer to video-visualization.mp4 for video visualizations of ActionVOS comparing to conventional RVOS settings.

## References

1. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: Proceedings of the European Conference on Computer Vision. pp. 720–736 (2018)
2. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. International Journal of Computer Vision pp. 1–23 (2022)
3. Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., Damen, D.: Epic-kitchens visor benchmark: Video segmentations and object relations. Advances in Neural Information Processing Systems **35**, 13745–13758 (2022)
4. Django Software Foundation: Django
5. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European Conference on Computer Vision. pp. 740–755 (2014)
6. Liu, C., Ding, H., Jiang, X.: Gres: Generalized referring expression segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23592–23601 (2023)
7. Tokmakov, P., Li, J., Gaidon, A.: Breaking the" object" in video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22836–22845 (2023)
8. Wu, J., Jiang, Y., Sun, P., Yuan, Z., Luo, P.: Language as queries for referring video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4974–4984 (2022)
9. Yu, J., Li, X., Zhao, X., Zhang, H., Wang, Y.X.: Video state-changing object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20439–20448 (2023)