Supplementary Material for Video State-Changing Object Segmentation

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
XMem [1]	66.7	59.7	73.7
XMem + NCC	69.9	62.7	77.1
XMem + Our FT	75.1	68.4	81.8

Table A. Quantitative comparison between Naive Cycle Consistency and our fine-tuning strategy. The performance of directly applying XMem or fine-tuning via the Naive Cycle Consistency is significantly worse than our strategy.

Method	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$cc\mathcal{J}$
VISOR-pretrained	52.8	45.8	59.7	41.6
Ours	76.5	70.0	83.1	64.7

Table B. Performance of ResNet-50 (pretrained on VISOR-VOS) on our VSCOS task.

In this supplementary material, we include (1) a **video demo** of our VSCOS results, and (2) this document containing additional details and explanations about our baseline approach.

A. Our Baseline Method

Hyperparameter Settings. A weight decay of 0.05 is applied, and we use a multistep learning rate schedule to reduce the learning rate to 1e-6 at 1,000 steps. For the EMA teacher model, we use a warmup mechanism such that the teacher-student loss is performed after 100 iterations, since both the teacher and student models perform poorly at first.

Implementation Details. In each iteration, we pick eight temporally ordered frames including the first and last frames from a training video clip. First, we use the first frame as a reference to predict the second frame's mask. Then we use the first and second frames as references to predict the third frame's mask. The memory bank is limited to at most 3 frames. So for the frames after the fourth frame, we will randomly choose three previously-occurring frames as references. In the inference stage, we use a three-level memory mechanism, following XMem[1].

Naive Cycle Consistency (NCC). To justify our fine-tuning strategy, we present results for a naive cycle consistency strategy. The model is provided with the first frame mask and propagates it forward to obtain a predicted last frame. Then we provide the predicted last frame mask as a reference and propagate it backward to predict the first frame mask. We apply a cross-entropy loss and a Dice segmentation loss only between the first frame mask ground truth and the predicted first frame. Additionally, we also apply the Mean Teacher losses described in the main paper. The results are shown in Table A. The performance of naive cycle consistency (NCC) is higher than XMem without fine-tuning, but significantly lower than our fine-tuning strategy. This is probably because in the NCC approach, the loss is only calculated for the first frame. Therefore, the model is not explicitly required to understand the state change of the object and the appearance of the object after the state change.

Optical Flow Model Details. Conceptually, in some types of state changes, optical flow provides additional motion information. To verify this, we applied K-Means clustering on the optical flow of our VSCOS dataset. As shown in Figure C, pixels corresponding to a falling piece of cucumber have consistent motion patterns, which is helpful for segmenting this piece. Inspired by this pilot study, we investigate an intuitive way to integrate optical flow with our baseline model as shown in Figure D. The optical flow is provided by the EPIC dataset [2] and we use a lightweight ResNet18 pretrained on ImageNet as the flow encoder. Following [4], we use 5 stacked optical flows as input and subtract the mean of optical flows to reduce the effect of global motion. We construct the Fusion Module with a Residual Block [3], a CBAM Block [5], and a Residual Block. This reduces the feature dimension to the same as the original XMem, and we do not change the XMem decoder structure.

Successful Cases of Our XMem-SC. Figure B shows representative cases where our XMem-SC is better than the baseline XMem.

Success Cases Where Integrating Optical Flow Brings Better Performance. Figure A shows representative cases where integrating optical flow to our XMem-SC is better than not using flow. Both are finetuned on our VSCOS dataset. We observe that using flow seems to be more successful in cases where there is a more pronounced object movement.

VISOR-pretrained results. In Table B we show empirically that VISOR-pretrained backbone fails for our VSCOS task, further proving that our task is challenging and not directly resolvable using the existing VISOR dataset.

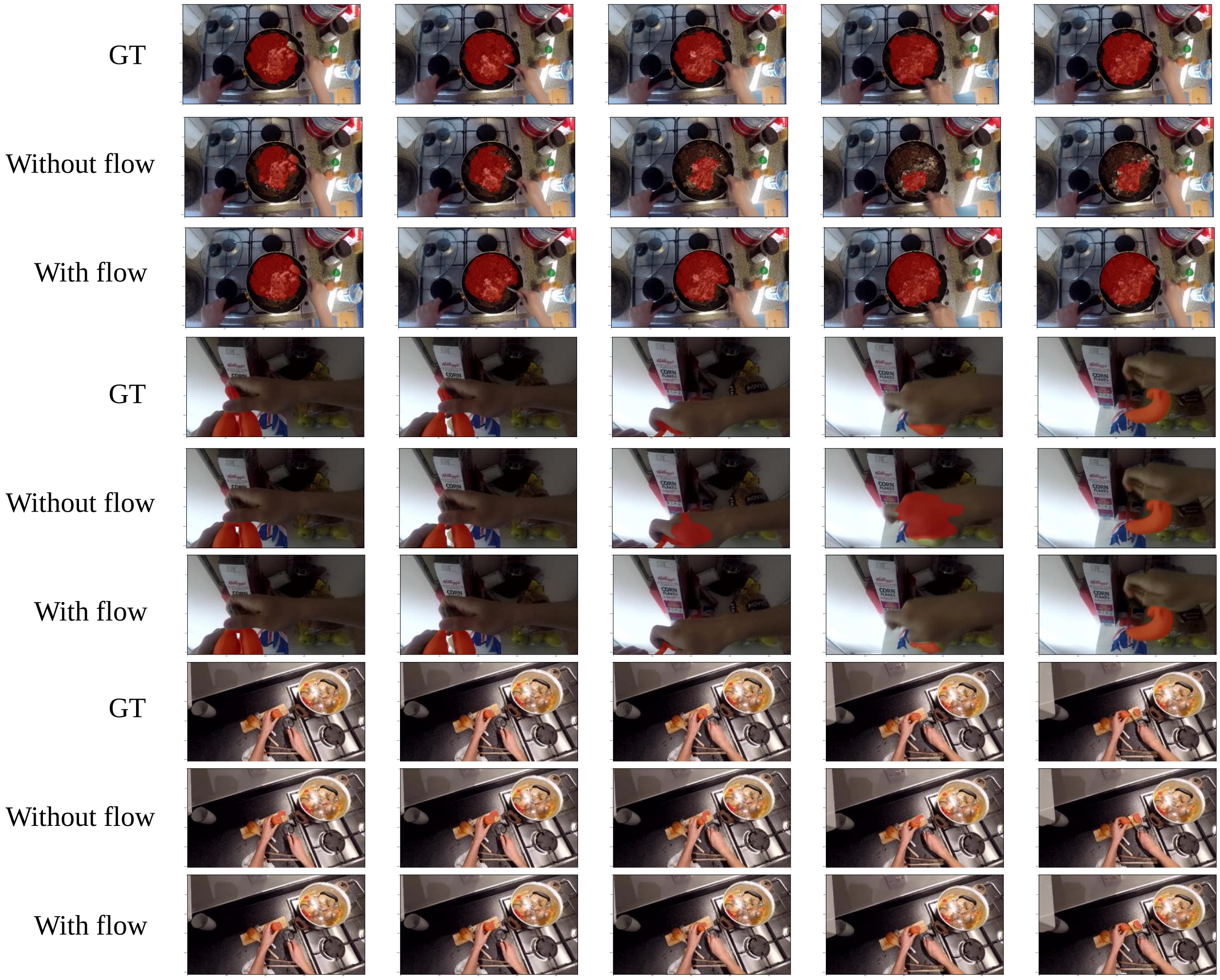

Figure A. Visualization of representative cases where integrating flow is better than not integrating flow. For each example, **top:** ground truth; **middle:** XMem-SC without flow; **bottom:** XMem-SC with the flow.

References

[1] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 1, 3

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[4] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NeurIPS*, 27, 2014. 1

[5] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 1

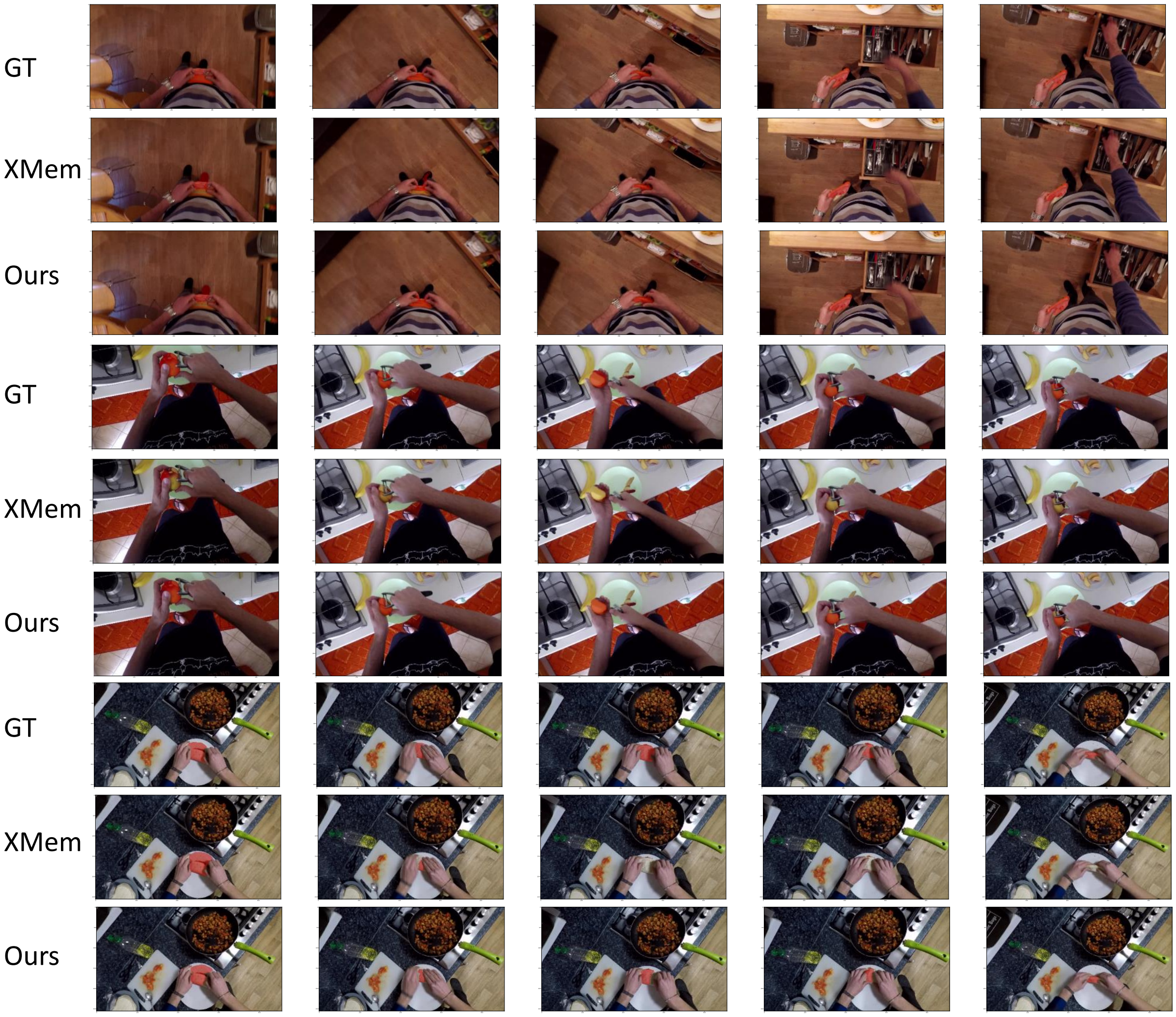

Figure B. Visualization of representative cases where our XMem-SC is better than XMem without finetuning. For each example, **top:** ground truth; **middle:** XMem-SC; **bottom:** XMem. **Best viewed in color with zoom.**

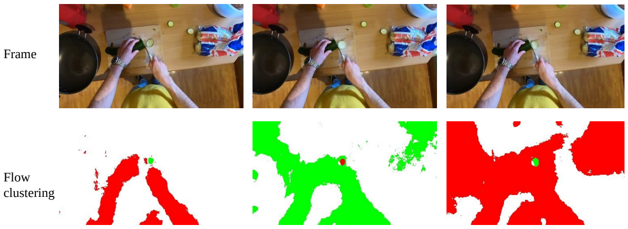

Figure C. Visualization of optical flow clustering. We directly use K-Means to cluster the optical flow. It can be seen that the falling pieces have a consistent pattern of motion. This observation motivates us to integrate motion information to address our VSCOS task.

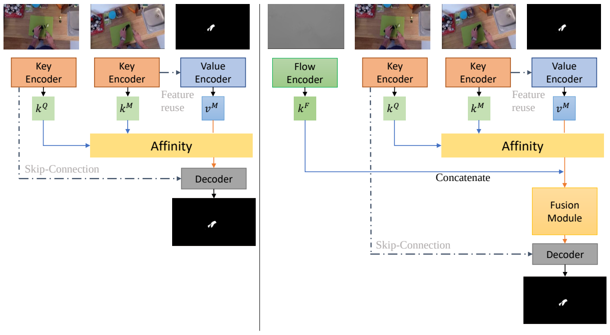

Figure D. **Left:** our baseline model XMem [1]. **Right**: our proposed model that integrates optical flow information.