Video State-Changing Object Segmentation

Jiangwei Yu[1*] Xiang Li[1*] Xinran Zhao[2] Hongming Zhang[3] Yu-Xiong Wang[1]

[1]University of Illinois at Urbana-Champaign
[2]Carnegie Mellon University [3]Tencent AI Lab, Bellevue
{jy79, xiangl12}@illinois.edu xinranz3@andrew.cmu.edu
hongmingzhang@global.tencent.com yxw@illinois.edu

Abstract

Daily objects commonly experience state changes. For example, slicing a cucumber changes its state from whole to sliced. Learning about object state changes in Video Object Segmentation (VOS) is crucial for understanding and interacting with the visual world. Conventional VOS benchmarks do not consider this challenging yet crucial problem. This paper makes a pioneering effort to introduce a weakly-supervised benchmark on Video State-Changing Object Segmentation (VSCOS). We construct our VSCOS benchmark by selecting state-changing videos from existing datasets. In advocate of an annotation-efficient approach towards state-changing object segmentation, we only annotate the first and last frames of training videos, which is different from conventional VOS. Notably, an open-vocabulary setting is included to evaluate the generalization to novel types of objects or state changes. We empirically illustrate that state-of-the-art VOS models struggle with state-changing objects and lose track after the state changes. We analyze the main difficulties of our VSCOS task and identify three technical improvements, namely, fine-tuning strategies, representation learning, and integrating motion information. Applying these improvements results in a strong baseline for segmenting state-changing objects consistently. Our benchmark and baseline methods are publicly available at https://github.com/venom12138/VSCOS.

1. Introduction

Object state changes are common in the real world. For example, when slicing a cucumber, the cucumber's state changes from *whole* to *sliced*. Humans learn commonsense knowledge about actions and associated objects by memorizing the state change in a certain time period [12]. Understanding state changes in visual perception tasks, for example, video object segmentation (VOS), is also crucial for autonomous agents to interact safely and efficiently with objects. In the example of slicing a cucumber, without state change knowledge, an autonomous agent might not know how to pick up and cut the cucumber such that it becomes slices. However, objects under state changes are largely ignored in previous VOS research. Existing VOS benchmarks tend to focus on normal objects, while overseeing the significantly more difficult state-changing ones with shifting appearances.

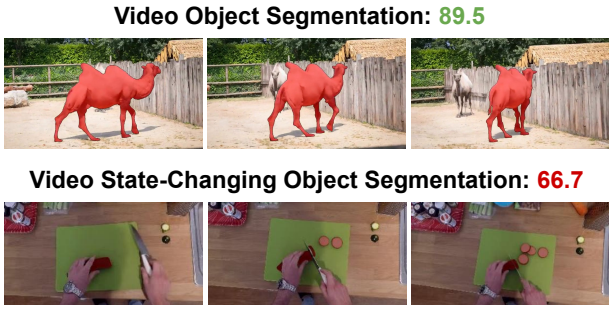

Figure 1. We propose Video State-Changing Object Segmentation (VSCOS), which is significantly more challenging than conventional VOS. State-of-the-art VOS model XMem [2] performance drops from 89.5 $\mathcal{J}\&\mathcal{F}$ (Jaccard & F-Score) to 66.7 $\mathcal{J}\&\mathcal{F}$ on our benchmark, because it fails to associate drastically changing object appearance. **Best viewed in color with zoom.**

This work investigates this under-explored problem of object state change in VOS. To the best of our knowledge, we are the first to formally define the task of Video State-Changing Object Segmentation (VSCOS). VSCOS aims to predict pixel-wise masks of state-changing objects in each frame of the video, given the first frame mask as reference.

In an effort to facilitate research on the VSCOS task, our first contribution is to construct a dedicated benchmark that reveals the failure of existing VOC methods and identifies

*Equal contribution.

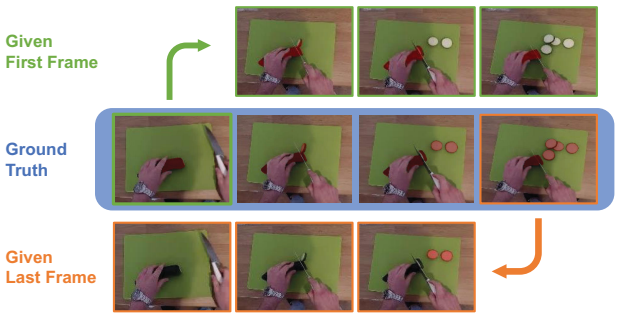

Figure 2. Visualization of our pilot study. On an example of cutting a cucumber, we apply a state-of-the-art VOS model XMem [2]. Provided with the first frame mask as a reference (**top row**), the model segments the *whole* cucumber but omits the *sliced* pieces. Similarly, provided with the last frame mask as a reference (**bottom row**), the model segments the *sliced* pieces but loses track of the *whole* part of the cucumber. This shows the state-of-the-art model segments the object in each individual state but fails to associate the segmentation when the state change happens. This key difficulty motivates us to derive the setting for our VSCOS task. **Best viewed in color with zoom.**

the key challenges of the VSCOS task. As shown in the example in Figure 1, on conventional VOS benchmarks (*e.g.*, DAVIS-2017 [11]), the model is expected to segment normal objects that do not experience major state or appearance changes like camels. Therefore, state-of-the-art VOS methods (*e.g.*, XMem [2] or DeAOT [23]) could satisfactorily segment these objects in videos, by matching the current frame's visual appearance to the reference frames or previous frames' predictions. By contrast, on our benchmark, the state-changing objects have large appearance changes. Given the first frame mask as reference, the model fails to coherently segment both the whole cucumber and the slices before and after the cutting action, which is notably more challenging. Correspondingly, the performance of the state-of-the-art VOS model XMem drops from 89.5 $\mathcal{J}\&\mathcal{F}$ on DAVIS-2017 to 66.7 $\mathcal{J}\&\mathcal{F}$ on our benchmark. This result highlights the significance of learning about object state changes in VOS.

We further show in Figure 2 that state-of-the-art VOS methods fail on our VSCOS benchmark, because they lose track of objects when state changes happen. This phenomenon indicates that the state-of-the-art VOS models lack the understanding of the identity of objects experiencing state changes. Therefore, we highlight a key difficulty of our VSCOS task as that the existing VOS models can segment the object in each individual state reasonably given the correct reference, *but cannot associate the segmentation before and after the state change*.

Our benchmark also possesses several desirable properties, as we have constructed it following *two* crucial principles. Primarily, our VSCOS benchmark should be established in an *annotation-efficient* fashion, so it could be easily extended to different video datasets. Doing so also encourages the advancement of weakly-supervised VSCOS methods. Based on the previous discovery that the key difficulty in VSCOS is the association, our setting provides two annotated frames for each training video, namely, the first and last frames. At test time, only the first frame mask is provided for online inference.

Another principle is that we propose an *open-vocabulary* setting alongside our conventional setting. This setting aims to test the models' generalization to novel state changes and objects that are previously unseen in training. Our open-vocabulary setting simulates a practical scenario where the trained model may encounter new types of objects under seen state changes, new types of state changes on seen objects, or even completely novel state changes and objects. This setting advocates models that do not overfit to categories seen in training, but generalize to the complex scenarios in the open world.

Based on our proposed VSCOS benchmark, we investigate how to adapt any existing VOS models to enable robust segmentation for state-changing objects, and propose our baseline method. Our baseline contains three components centered around solving the key difficulty of segmentation association before and after the state changes. *First*, we design an effective fine-tuning method that explicitly tackles the association problem with cycle consistency and a teacher-student loss. Our fine-tuning strategy significantly improves VSCOS performance, while avoiding training instability and trivial solutions. *Then*, we point out a promising direction in improving feature representation for VSCOS. Specifically, the features for the object region before and after the state changes should be aligned, while both should be distinguished from the background feature. As an initial approach, we adapt Contrastive Random Walk [7] to be an auxiliary loss and it demonstrates a noticeable performance improvement. *Finally*, we explore whether motion information in the form of optical flow could assist VSCOS in connecting the states before and after the changes. We design a simple approach to fuse flow features into VOS models and also observe a minor improvement. Here we do not claim that our baseline method is necessarily an optimal strategy, but it points out key research directions for VSCOS including fine-tuning, feature learning, and integrating motion information.

We further analyze the results of our baseline method on VSCOS from different perspectives. For example, we investigate the contribution of different design decisions, the performance comparison for different action categories, as well as the different phenomena in different sets of the open-vocabulary setting. From these experiments, we draw empirical conclusions on how to improve VSCOS performance. Finally, we observe and categorize key failure cases of our baseline model and the main difficulties of our VSCOS task.

Dataset	Perspective	Annotated Length (h)	# of Action Categories	Segmentation Label	Open-Vocabulary Setting
[1]	Third-person	1	7	✗	✗
[8]	Egocentric	3	14	✗	✗
ChangeIt [15]	Third-person	48	44	✗	✗
VSCOS (Ours)	Egocentric	4	271	✓	✓

Table 1. Comparison of dataset statistics between our VSCOS benchmark and previous work. Our VSCOS benchmark features the most varied action categories, as well as fine-grained segmentation labels and open-vocabulary settings.

We also discuss the limitations of our approach and future work.

To summarize, our contributions are three-fold:

(1) We propose a crucial yet under-investigated problem of Video State-Changing Object Segmentation (VSCOS).

(2) We annotate a state-changing VOS dataset and build our VSCOS benchmark. Our benchmark is annotation efficient and contains an open-vocabulary setting to evaluate the model's generalization capability.

(3) We identify the key difficulty of our task: associating the object segmentation before and after state changes. Based on this observation, we establish a model-agnostic baseline method that adapts existing VOS models for VSCOS. Our baseline method points out key research directions for the VSCOS task. We present and analyze the baseline results, as well as the key challenges of our benchmark.

2. Related Work

Video Object Segmentation. Video object segmentation (VOS) has been comprehensively studied by previous work [17]. On currently widely-used VOS benchmarks, *e.g.*, DAVIS [11] and YouTube-VOS [18], the object of interest does not experience large appearance or state changes. Therefore, representative recent methods [10, 20, 2, 13, 21, 23] mostly model the VOS problem as matching object appearances between query frames and reference frames, leading to their failure in the challenging object state change setting where drastic appearance changes happen. We aim to bridge this gap by proposing the video state-changing object segmentation (VSCOS) task, which requires an understanding of object identity through state changes beyond appearance matching.

Object State Changes in Videos. Object state changes have been investigated in the context of videos. The task of jointly discovering states and actions by utilizing the causal relationship between them is investigated [1]. States and actions have also been studied under the name of *fluents* and *tasks* [8], where they are classified in a closed world setting via beam search. Recently, a self-supervised method has been proposed to jointly localize action and state changes temporally from noisy untrimmed long videos [15]. More comprehensive tasks of Point-of-No-Return detection and state-changing object detection have been proposed in Ego4D [5].

We develop upon these previous efforts by proposing a *more fine-grained* video object segmentation benchmark for spatial and temporal state change knowledge. We task the model with pixel-wise segmentation of state-changing objects coherently for each frame of the video, while previous methods only focus on frame-level classification or temporal detection. Meanwhile, prior work leverages the causal relationship between state and action, while our VSCOS benchmark does not have such a strong dependency. This allows us to design an open-vocabulary setting to test the generalization to unseen objects and novel types of state changes.

Notably, a contemporary work [16] investigates video object segmentation under transformations. Compared with [16], our effort includes different types of state changes. Meanwhile, we advocate a more challenging weakly-supervised setting, where we only adapt models on state-change videos that we annotated with first and last frame masks. This provides new opportunities for investigating this practical and annotation-efficient scenario.

3. Video State-Changing Object Segmentation

In this section, we describe our formulation of the Video State-Changing Object Segmentation (VSCOS) task. Based on a pilot study on representative data, we identify the key difficulty of VSCOS: state-of-the-art VOS models could segment the object in its initial state, but they are unable to associate the same object before and after the state change to obtain the mask for the final state. This observation motivates the design of our benchmark.

Data. We choose a representative scenario of egocentric videos from EPIC-Kitchens [3]. The dataset is collected by volunteers wearing GoPro cameras on their heads during cooking. Cooking videos provide a rich set of state-changing objects due to frequent human-object interaction. Therefore, it is a natural testbed for modeling state-changing objects in VOS. Meanwhile, these egocentric videos are continuously captured. This avoids the problem of jump cuts and edits in third-person state change videos [1, 15] that are problematic for VSCOS.

Pilot Study on Design of VSCOS. We evaluate state-of-the-art VOS models on our representative videos to investigate whether they could handle state-changing objects.

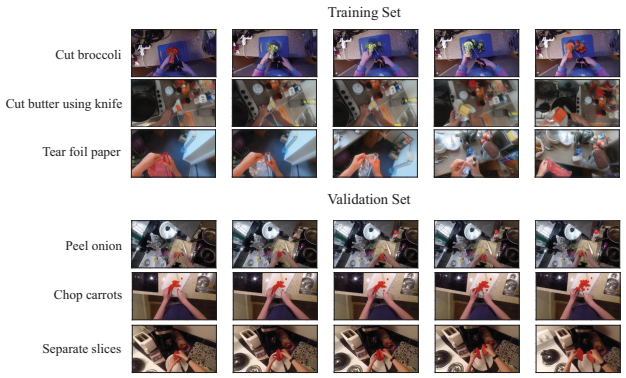

Figure 3. Visualization of data samples from our VSCOS benchmark. In the training set, only the first and last frames are annotated. **Best viewed in color with zoom.**

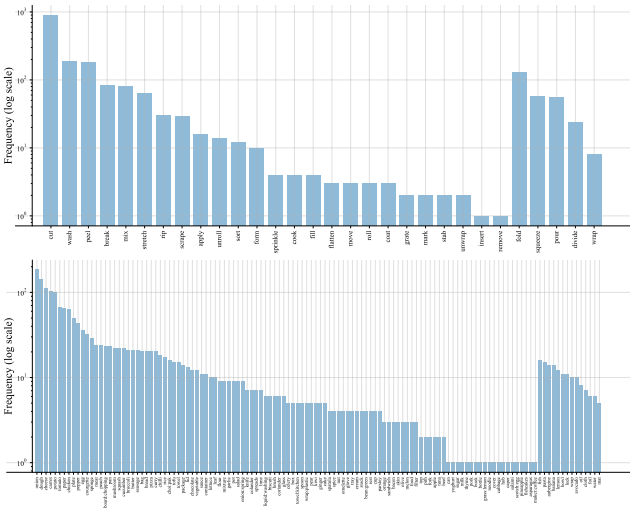

Figure 4. Statistics of the action (**top**) and object (**bottom**) categories in our VSCOS dataset. Our dataset contains a large number of varied actions and objects. The number of examples in these categories follows a *long-tail* distribution, so we visualize them in log scale. We separate seen categories (**left**) from novel categories (**right**).

We focus on the semi-supervised VOS setting, where the model takes the segmentation mask of the first frame in the video and propagates the segmentation mask across the entire video. We discover that the state-of-the-art model suffers a large performance drop on videos with object state changes. To provide insight into why they fail, we visualize the segmentation results of a representative state-of-the-art model XMem [2] in Figure 2. On the top row, we simulate an online inference scenario and provide the mask of the first frame of the whole cucumber before the state change. The model segments the rest of the whole cucumber reasonably well, but loses track of the slices after the state change occurs. This shows that the state-of-the-art VOS model is insufficient for obtaining persistent segmentation through state changes, potentially due to the large appearance change.

Similarly, on the bottom row, we reverse the video and provide the model with the mask of the original last frame, containing mostly cucumber slices. In this case, the model segments the slices well, but loses track of the whole cucumber. This phenomenon is prevalent in the results of our pilot study. It points out the key difficulty of segmenting state-changing objects: The state-of-the-art VOS model can segment objects in their initial state, but cannot continue to segment the object in their final state after the change. In other words, the model cannot robustly associate the segmentation for different states through the state change.

VSCOS Task. We propose an annotation-efficient setting for our VSCOS task, such that it could be easily extended to other datasets. In this task, we aim to adapt a pretrained VOS model such that it could robustly segment objects undergoing state changes. As shown in the pilot study, the first and last frames provide reasonably accurate mask propagation for the part of objects in the initial and final states respectively. The state-of-the-art VOS model only fails through the state changes when the object's appearance transitions. Therefore, an annotation-efficient setting would be to provide the model with the ground truth mask for the first and last frames in training. At test time, we provide only the first frame annotation to the model for online inference.

To summarize, the model is supplied with short video clips containing state-changing objects in the training stage. The object's segmentation mask is provided for the first and last frames. This informs the model of the object's appearance before and after the state change. The model learns to propagate and associate the segmentation through the state change. During inference, the model is provided with the mask label of the first frame only. It should propagate this mask to all following frames in an online fashion robustly through the state change. Different from training, the model does not explicitly know the object's appearance in the final state and must infer it based on the video content and the learned knowledge about the state change in training.

Open-Vocabulary Setting. In the real world, a model will often encounter novel objects or even novel state changes unseen in training. To evaluate the generalization of VSCOS models in the open world, we construct an open-vocabulary setting alongside the conventional setting. In this open vocabulary setting, apart from the object and state-change pair seen in training, there are three additional scenarios. There are novel objects experiencing seen state changes, seen objects experiencing novel state changes, and the most difficult case where both the object and the state change are unseen in training. We report the performance for these four scenarios individually in our experiments.

Evaluation Metrics. Following previous work in VOS [17], we mainly evaluate VSCOS performance by Jaccard index $\mathcal{J}$, boundary accuracy $\mathcal{F}$, and their mean $\mathcal{J}\&\mathcal{F}$.

In addition, we propose a new metric specific to VSCOS, named *connected component Jaccard index*, or $cc\mathcal{J}$. In

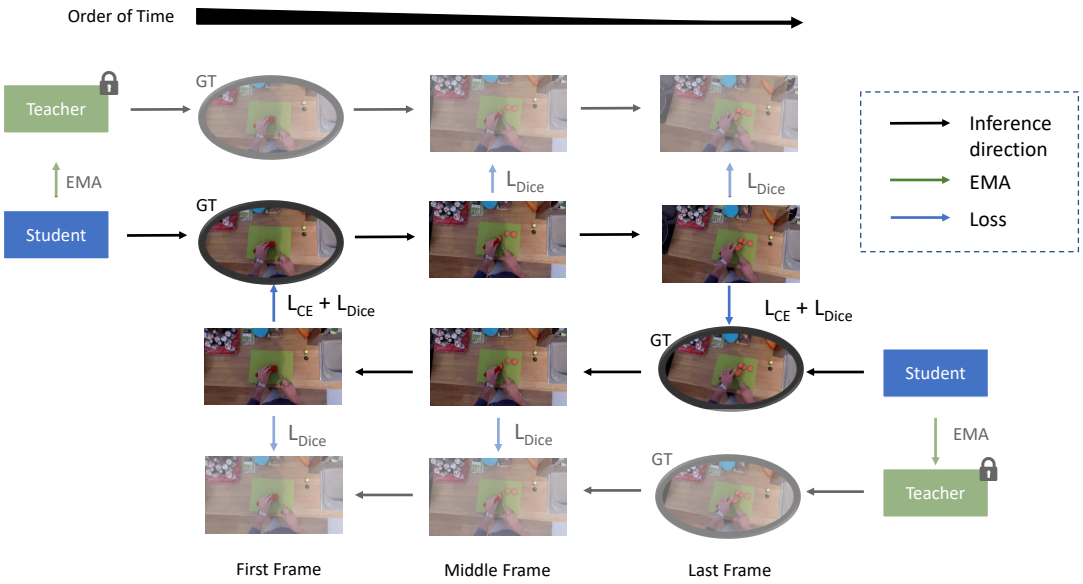

Figure 5. Our proposed baseline fine-tuning strategy. The oval masks represent ground truth annotations, while the square masks represent model predictions for each frame. At each step, both the student and teacher models perform inference from either direction and align the predicted mask with the ground truth. And the student predictions are aligned with the teacher's predictions correspondingly. The teacher model is an EMA copy of the student model and is not updated by gradients.

VSCOS, objects may often become fragmented, resulting in multiple small pieces. If a model only focuses on the primary part of the object and overlooks the small pieces, it could still obtain a high $\mathcal{J}\&\mathcal{F}$. To address this, we calculate Jaccard index based on each connected component and take an average over all components. Specifically, we first find all connected components in the ground truth mask and the model output mask. Then, we find an optimal bipartite matching that maximizes the average Jaccard index over all matches. We take that averaged result as our $cc\mathcal{J}$. By quantifying the performance for each connected component, this $cc\mathcal{J}$ metric better captures the ability of the model to robustly segment objects through state changes, particularly those that involve fragmentation.

4. Benchmark

Annotation. Contemporary to our work, EPIC-Kitchens has released mask labels [4]. Unfortunately, their human-annotated labels are too sparse to be useful for our fine-grained VSCOS task, while their machine-propagated dense labels are often not accurate. So we do not apply them, and we annotate our segmentation masks instead. We first manually filter the action labels to extract a subset of EPIC-Kitchens that contains state changes. After obtaining this set of state change videos, we annotate the segmentation labels.

As described in the task setting, we annotate the first and last frames of training videos. For the test videos, we annotate densely (one frame per second) to guarantee a reliable evaluation. Segmentation labeling is known to be labor intensive, since it requires drawing pixel-wise masks around the spatial extent of objects. This challenge is more severe on our dataset, since the state changes involve many fine details and small pieces. To further alleviate annotation burdens, we modify f-BRS [14] interactive segmentation GUI as our interactive annotation tool, which proposes regions based on user clicks. We manually double-check the annotations and fix ambiguities to ensure the reliability of our segmentation labels. For a specific video, we annotate the manipulated object based on the action label. For example, if the video action is *cutting cucumber*, we annotate the cucumber being cut.

Statistics and Comparison. There are 1,905 video clips containing state changes in our VSCOS dataset, each spanning 7.4 seconds on average. These videos span 30 action categories and 124 object categories, yielding a total of 271 valid combinations. We visualize these data samples in Figure 3. For the training set, we have 1,809 videos and 3,618 annotated frames in total. The average length of training videos is 442 frames. For the test set, we have 98 videos and 1,254 annotated frames. The average length of validation videos is 450 frames. We summarize that in our benchmark, there are four prominent categories of state changes:

1. Rigid Object Composition and Decomposition (combine, cut, split, disintegrate, unpackage, ...)
2. Non-rigid Object Transformation (pour (liquid), crack (egg))
3. Object Appearance Change (cook, clean, ...)
4. Object Articulation (open, close, twist, ...)

We compare our VSCOS dataset with previous efforts in Table 1. Our dataset has the most variety in action categories,

Method	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$cc\mathcal{J}\uparrow$
CFBI [19]	51.5	47.0	56.0	43.7
CFBI+ [22]	58.2	53.3	63.1	48.2
XMem [2]	66.7	59.7	73.7	54.8
AOT [21]	72.6	65.2	80.1	60.8
DeAOT [23]	73.3	65.6	80.9	60.7
XMem-SC (Ours)	76.5	**70.0**	83.1	64.7
DeAOT-SC (Ours)	**77.1**	69.7	**84.4**	**66.0**

Table 2. Results on the conventional setting of our VSCOS task. Our adapted VOS approaches achieve large performance improvements on our benchmark in all metrics. Note that $cc\mathcal{J}$ represents our proposed connected component Jaccard index in Section 3.

corresponding to a more varied set of state changes. Meanwhile, our dataset is the only existing work with pixel-wise segmentation labels and open-vocabulary settings.

5. Method

In this section, we investigate several different perspectives for adapting existing VOS models for the VSCOS task and we propose a baseline method. To tackle the key difficulty of associating the segmentation before and after the state change, we first propose an effective fine-tuning strategy for VOS models. Then we highlight the importance of improving the representation learning for VSCOS by aligning the region features before and after the state change. Finally, we attempt to integrate motion information in the form of optical flow to assist VSCOS. These improvements are *mostly agnostic* to the underlying VOS model. Here we take XMem [2] and DeAOT [23] as representative examples because of their strong performance.

Fine-tuning Strategy. In our annotation-efficient VSCOS setting, only the mask labels for the first and last frames is available for each video. Therefore, conventional VOS training is not feasible. The model is required to robustly associate the segmentation before and after the state change given the masks for the first and last frames. Therefore, we propose a natural cycle consistency approach. Specifically, we provide the first frame mask to the model to predict the last frame mask and calculate a loss between the result and the last frame ground truth. Then, in an opposing fashion, we provide the last frame mask to the model and predict the first frame mask. The loss is calculated again between the first frame result and the first frame ground truth. Here we follow XMem [2] or DeAOT [23] and use their respective losses.

Empirically, we discover that this fine-tuning strategy is effective but sometimes unstable in training. This is because of a trivial solution where the model predicts no mask for all middle frames and only memorizes the first and last frame masks. We alleviate this problem by introducing a teacher-student loss based on an exponential moving average (EMA) teacher model. The teacher and student models are identical, and both are initialized with a pretrained state-of-the-art VOS model (*e.g.*, XMem [2] or DeAOT [23]). During training, our teacher model is updated solely through the EMA of the student model's parameters and not updated by gradients. Apart from the aforementioned first and last frame loss, we apply a Dice segmentation loss for each training frame between the student and teacher models' predicted masks. This teacher-student loss alleviates the trivial solution by smoothing the training process, avoiding the abrupt changes in model update that leads to the trivial solution. We visualize our final fine-tuning strategy in Figure 5.

Representation Learning. Associating the segmentation before and after the state change requires special properties of the deep representation. Specifically, the feature of the object region before and after the state change should be aligned, while both should be distinct from the background feature. To show the promise of such desirable feature representation, we propose our baseline approach by adapting Contrastive Random Walk (CRW) [7] to be an auxiliary loss in fine-tuning. CRW is a self-supervised approach to align the features for corresponding regions in a video. This is accomplished by constructing a palindrome space-time graph from the video and performing link prediction in the graph. Instead of using image patches, we take the features from the image encoder of the VOS model and apply the CRW loss on each of our training frames.

Motion Information. During object state changes, the visual appearance changes drastically. However, in some types of state changes, the motion provides useful information to the association throughout the state change. For example, when slicing a cucumber and a small slice falls off, all pixels corresponding to the slice will have consistent movement when it is separated from the rest of the cucumber. This could potentially enable the model to associate the slice with the rest of the cucumber. Therefore, we make an initial attempt to introduce motion information in the form of optical flow into VOS models. Namely, we randomly initialize a ResNet [6] model as the flow encoder. The optical flow extracted with existing models is passed through the flow encoder to obtain flow features. We concatenate the flow feature with the appearance feature from the VOS model, and apply a light-weight fusion module to combine them. Finally, we feed the fused flow and appearance feature into the decoder. The flow encoder and fusion module are trained end-to-end during our fine-tuning.

Loss Function. Suppose we sample n frames from the video for training, including the first frame with label y_0 and the last frame with label y_n. We represent our student model as ϕ^S and our teacher model as ϕ^T, and we use λ as the decay parameter in EMA. α and β are loss balancing factors. We use subscripts to denote the frame index. Then we derive our loss, taking XMem [2] as an example as follows:

Method	Seen State Changes + Seen Objects	Seen State Changes + Novel Objects	Novel State Changes + Seen Objects	Novel State Changes + Novel Objects
XMem [2]	68.7	68.7	63.0	57.4
XMem-SC (Ours)	**77.5**	**77.2**	**64.1**	**66.2**

Table 3. Open-vocabulary results on our VSCOS benchmark measured by the $\mathcal{J}\&\mathcal{F}$ score. All four sets are challenging to different degrees for XMem. The performance improves consistently for all sets for our XMem-SC.

FT	TS	CRW	OF	$\mathcal{J}\&\mathcal{F}\uparrow$	$\mathcal{J}\uparrow$	$\mathcal{F}\uparrow$	$cc\mathcal{J}\uparrow$
				66.7	59.7	73.7	54.8
✓				73.6	66.6	80.6	61.6
✓	✓			75.1	68.4	81.8	62.9
✓	✓		✓	75.4	68.6	82.3	62.9
✓	✓	✓	✓	76.5	70.0	83.1	64.7

Table 4. Ablation study of our XMem-SC. "FT" refers to our fine-tuning strategy without the teacher-student loss. "TS" refers to teacher-student loss. "CRW" refers to the Contrastive Random Walk loss. "OF" refers to the integration of optical flow. We show that all these strategies improve the performance to different extents.

$$
\begin{aligned}
L_{\text{XMem-SC}} =& L_{\text{CE}}(\phi^S(y_0)_n, y_n) + L_{\text{CE}}(\phi^S(y_n)_0, y_0) \\
&+ L_{\text{Dice}}(\phi^S(y_0)_n, y_n) + L_{\text{Dice}}(\phi^S(y_n)_0, y_0) \\
&+ \alpha(\sum_{i=1}^{n} L_{\text{Dice}}(\phi^S(y_0)_i, \phi^T(y_0)_i)) \\
&+ \alpha(\sum_{i=0}^{n-1} L_{\text{Dice}}(\phi^S(y_n)_i, \phi^T(y_n)_i)) \\
&+ \beta L_{\text{CRW}}
\end{aligned}
$$

where $\phi^T \leftarrow \lambda\phi^T + (1-\lambda)\phi^S$.

Implementation Details. We train our student model with the AdamW [9] optimizer. We train for 10000 iterations on our VSCOS dataset with a base learning rate of 1e-5. 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 1000 steps. We sample $n = 8$ frames from each video with a batch size of 8 videos. For the EMA teacher model, we set the student-teacher loss weight α to 0.01, CRW loss weight β to 1, and the decay parameter λ to 0.99. Since we only aim to stabilize training with our mean teacher, we differ from previous semi-supervised learning works and do not apply different augmentations for student and teacher input videos.

When calculating the CRW loss, instead of image patches, we use the features from the image encoder of the VOS model. We apply average pooling such that its spatial dimension is 12×12. As for the optical flow, we concatenate 5 adjacent flow frames, normalize these frames, and feed them to a randomly initialized ResNet-18 [6] flow encoder

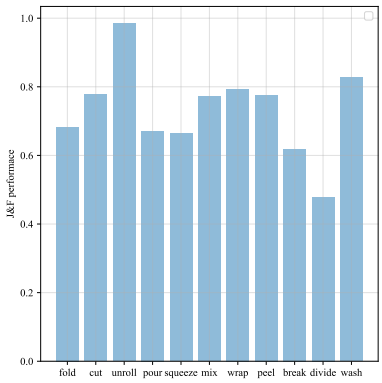

Figure 6. Average performance of our XMem-SC model for each action category in our VSCOS benchmark. The performance is measured by $\mathcal{J}\&\mathcal{F}$. Different action classes have different difficulties for VSCOS.

to obtain the flow feature. Finally, we concatenate the flow feature with the corresponding image feature and use a light-weight multilayer perceptron(MLP) fusion model to produce the final feature used to predict the mask. Additional hyperparameter settings are discussed in the supplementary material.

6. Experiments

In this section, we introduce our baseline results on the VSCOS benchmark.

Main Results. We evaluate several widely used VOS models on our VSCOS benchmark,including CFBI [19], CFBI+ [22], XMem [2], AOT [21], and DeAOT [23]. We pick the best performing and most representative method XMem [2] and DeAOT [23] and build our baseline method XMem-StateChange (XMem-SC) and DeAOT-StateChange (DeAOT-SC) based on them as described in Section 5. Table 2 shows the performance comparison of several baseline methods. These VOS models are pretrained on DAVIS-17. Intriguingly, we observe that *the relative performance of different methods is inconsistent with conventional benchmarks without state change*. For example, while AOT [21] performs slightly worse than XMem [2] on DAVIS-17 (84.9 vs. 86.2 $\mathcal{J}\&\mathcal{F}$), it performs noticeably better than XMem on our benchmark (72.6 vs. 66.7 $\mathcal{J}\&\mathcal{F}$). This observation again highlights the importance of our dataset for evaluating the under-explored problem of VOS with state-changing ob-

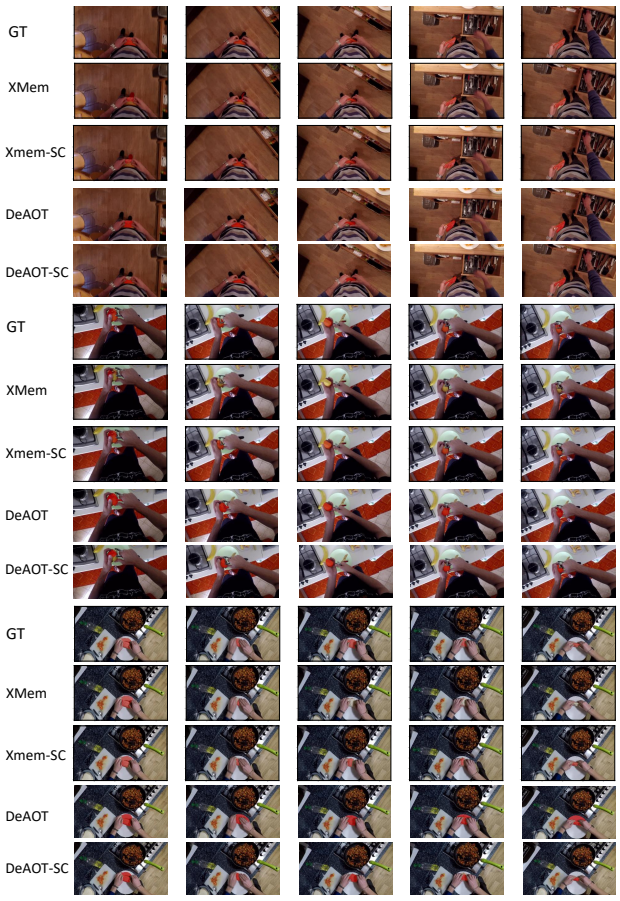

Figure 7. Visualization of qualitative results of our method. **Best viewed in color with zoom.**

jects, as an advanced VOS method does not necessarily lead to improved performance on dealing with more challenging state-changing objects.

We empirically demonstrate that after integrating our adaptation, the performance achieves a large improvement. For example, for XMem-SC, the improvement is 9.8 $\mathcal{J}\&\mathcal{F}$, 10.3 $\mathcal{J}$, 9.4 $\mathcal{F}$, and 9.9 $cc\mathcal{J}$. Meanwhile, for the stronger baseline DeAOT, our adaptation still manages to improve for 3.8 $\mathcal{J}\&\mathcal{F}$, 4.1 $\mathcal{J}$, 3.5 $\mathcal{F}$, and 5.3 $cc\mathcal{J}$. Qualitatively, we show in Figure 7 that our adapted models no longer lose track of the state-changing objects and tend to achieve a consistent segmentation through state changes.

Ablation Study. In Table 4, we show how different design decisions have an impact on the performance of our XMem-SC model on our VSCOS benchmark. If we fine-tune the model without the teacher-student loss, the training becomes unstable and sometimes collapses to a trivial solution. In cases where the model is trained well, we already achieve a relative improvement of 6.9 $\mathcal{J}\&\mathcal{F}$. By adding the teacher-student loss, we stabilize training and further improve the $\mathcal{J}\&\mathcal{F}$ by 1.5. This shows that our proposed fine-tuning strategy is effective. Our attempt of integrating motion information by introducing optical flow does bring a marginal performance improvement of 0.3 $\mathcal{J}\&\mathcal{F}$, and improves $\mathcal{F}$ slightly more by 0.5. Since our most direct way of integrating optical flow improves performance, we argue that integrating motion information in more advanced manners is a promising direction for VSCOS. Finally, our Contrastive Random Walk auxiliary loss improves the performance moderately by 1.1 $\mathcal{J}\&\mathcal{F}$, and especially improves $cc\mathcal{J}$ by 1.8. This shows the promise of aligning the local feature representation of the objects before and after the state change, while distinguishing their features from the background. More sophisticated strategies may bring even larger performance improvements in this regard.

Performance Breakdown by Action. We examine the performance of our best models on VSCOS by action categories and compare them in Figure 6. The bar chart shows the per-action VSCOS performance of XMem-SC. Note that each action corresponds to a type of state change. It is observed that some actions that are most difficult for VSCOS (*e.g.*, pour, squeeze, and break) have large temporal appearance changes. Conversely, actions with more minor or intricate temporal appearance changes (*e.g.*, wrap and wash) are reasonably easy for VSCOS.

Open-vocabulary Setting. Table 3 shows the performance of XMem and XMem-SC in the open-vocabulary setting of our VSCOS benchmark. All splits pose a degree of challenge for XMem. After integrating our three strategies, the performance consistently improves across all four splits, though the degree differs. This shows that our XMem-SC learns a somewhat generalizable concept of state change that enhances model performance on both seen and unseen state changes and objects. Interestingly, the improvement on *novel state changes + seen objects* is less than the other splits. This potentially suggests that a seen object category does not necessarily make the VSCOS problem easier. The model may overfit by memorizing the potential change of state by the object's appearance before the state change. When a novel state change occurs on a seen object category, the model might still be overfitted to the seen state change. This may result in the model experiencing more difficulty learning this novel state change than never having seen the object. This could be a potential difficulty for our VSCOS task.

7. Discussion

Failure Modes of Our Approach. Figure 8 shows cases where our XMem-SC model fails on our VSCOS dataset. We identify three main failure modes, which may indicate the difficulties of our benchmark:

(1) Over-segmentation. The first example shows that the model tends to over-segment the background more than it under-segments, which has been observed in a number of

Figure 8. Visualization of our three main failure cases. The three examples show over-segmentation, missing parts, and difficult state change cases respectively. The top result for each example is the ground truth, while the bottom result is from our XMem-SC. **Best viewed in color with zoom.**

different cases on our benchmark. A potential explanation is that in training, the model only receives direct supervision at the first and last frames. In the middle frame, there is no supervision apart from the mean teacher loss. This potentially means that the model could generate an arbitrary mask in the middle frames as long as it could correctly propagate the first frame label through the middle frames to the last frame and vice versa. In this situation, over-segmenting might be easier for propagating labels than under-segmenting. We assume this might be the reason for over-segmentation, when the model is uncertain in the middle frames of certain videos.

(2) Missing parts. Baseline XMem tends to lose track of the separated parts of state-changing objects (*e.g.*, the pieces of a cucumber being sliced off). Our XMem-SC improves in most cases, but still misses these parts in some difficult cases. The second example depicts this phenomenon in a video where the lighting is dim and the scene is cluttered. Although these cases are in the minority, it still calls for better future methods for adapting VOS models.

(3) Difficult state change cases. Since our VSCOS includes a large range of different state changes, some are more difficult for the model to identify. The third example shows a tablecloth being squeezed, where the object of interest experiences shape changes, gets occluded by human hands, and shrinks severely in spatial extent. Such cases are inherently difficult and our XMem-SC does not perform well on these especially difficult samples, here our model considers a part of the hand also as the object of interest.

Difference to VISOR. We build our dataset based on a representative scenario of egocentric videos from EPIC-Kitchens, where understanding state change is crucial. A recent dataset VISOR [4] is also built on EPIC-Kitchens with segmentation annotations. However, we do not utilize VISOR annotation when building our dataset due to its sparsity of human annotations. Specifically, at the start and end of the state changes, there is usually no human annotation from VISOR. At these crucial times, the machine-propagated labels are often not accurate enough for VSCOS. In the supplementary material, we show empirically that VISOR-pretrained backbone fails for our VSCOS task, further proving that our task is challenging and not directly resolvable by using the existing VISOR dataset.

Limitations and Future Work. Finally, we list some limitations of our approach and future directions. Our VSCOS benchmark utilizes EPIC-Kitchen data as a representative scenario. We design the VSCOS benchmark in an annotation-efficient way such that it could be easily extended to other sources of video data. In the future, we plan to extend our benchmark to more in-the-wild video datasets, *e.g.*, Ego4D [5]. We also plan to investigate VSCOS in more depth, for example, in scenarios with multiple objects and long-form state changes. Meanwhile, our baseline models achieve reasonable performance improvements, but we do not claim that they are the optimal strategies. They serve as initial attempts to explore whether these directions of improving VOS models are effective for VSCOS. We advocate more advanced strategies for fine-tuning VOS models, improving feature representation, and integrating motion information.

8. Conclusion

We propose a challenging yet under-investigated problem of Video State-Changing Object Segmentation (VSCOS), where we evaluate the robustness of a VOS model for state-changing objects. We facilitate the research of this problem with a novel benchmark based on Egocentric video datasets in an annotation-efficient and open-vocabulary setting. We observe that this task is significantly challenging for existing VOS models, and identify the key difficulty in VSCOS being associating the object segmentation before and after the state change. Based on this observation, we investigate three main approaches to adapt existing VOS models and enable robust segmentation under state change. Namely, we explore effective fine-tuning strategies, representation learning, and integration of motion information. We combine them as our baseline method, and empirically show a large performance improvement. We hope our released benchmark could facilitate future research on the fine-grained understanding of object state changes.

Acknowledgement. This work was supported in part by NSF Grant 2106825, NIFA Award 2020-67021-32799, the Jump ARCHES endowment, the NCSA Fellows program, the Illinois-Insper Partnership, and the Amazon Research Award. This work used NVIDIA GPUs at NCSA Delta through allocations CIS220014 and CIS230012 from the ACCESS program.

References

[1] Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, 2017. 3

[2] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 1, 2, 3, 4, 6, 7

[3] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. *IJCV*, 2022. 3

[4] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR Benchmark: Video segmentations and object relations. In *NeurIPS Track on Datasets and Benchmarks*, 2022. 5, 9

[5] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *CVPR*, 2022. 3, 9

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[7] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *NeurIPS*, 2020. 2, 6

[8] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly recognizing object fluents and tasks in egocentric videos. In *ICCV*, 2017. 3

[9] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. In *ICLR*, 2019. 7

[10] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 3

[11] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv*, 2017. 2, 3

[12] Roger C Schank and Robert P Abelson. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press, 2013. 1

[13] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *ICCV*, 2021. 3

[14] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-BRS: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, 2020. 5

[15] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *CVPR*, 2022. 3

[16] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *CVPR*, 2023. 3

[17] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. A survey on deep learning technique for video segmentation. *TPAMI*, 2023. 3, 4

[18] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-VOS: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 3

[19] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 6, 7

[20] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 3

[21] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 3, 6, 7

[22] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 2021. 6, 7

[23] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *NeurIPS*, 2022. 2, 3, 6, 7