Appendix for Breaking the “Object” in Video Object Segmentation

Pavel Tokmakov Jie Li Adrien Gaidon
Toyota Research Institute
first.last@tri.global

In this appendix, we provide additional visualizations, datasets statistics and implementation details that were not included in the main paper due to space limitations. We begin by describing the contents of the supplementary video, which includes full versions of Figures 2 and 6 from the main paper in Section 1. We then provide further details on collecting VOST in Section 2 and report additional dataset statistics in Section 3. An enlarged version of Figure 3 from the main paper is shown in Figures 6 and 7. A discussion on limitations of annotation interpolation from [5, 13] in challenging scenarios can be found in Section 4. Finally, we provide the details of our proposed recurrent transformer module in Section 5 and further implementation details in Section 6.

1. Video Description

1.1. Annotation visualization

We begin by visualizing VOST annotations for several representative sequences from Figure 2 in the main paper in this video. Mask colours indicate instance ids, with grey representing ignored regions.

00:00-00:18 In the first sequence we can see 6 separate instance of corn being cut. According to our labeling strategy (see Section 2.2), only the instances that are being manipulated are labeled in the video. As the objects are separated into many parts and moved around the board, all the parts maintain the identity of the instance they originated from. We can also see that even the smallest parts are labeled with accurate masks.

00:19-00:54 Next video illustrates the broad semantic meaning of cutting. This sample of paper cutting in the context of making art features many small, thin regions that are accurately labeled, as well as an example of fast motion, when the object is separated into two parts of different colour towards the end of the clip.

00:55-01:15 In this outdoor video a piece of clay is being molded into a brick. In the process, it experiences major shape changes combined with a full occlusion. Moreover, the motions are fast resulting in a significant amount of blur. The corresponding regions are labeled as “Ignore” (shown in gray) to avoid ambiguity during training and evaluation.

01:16-01:40 Finally, the very challenging egg breaking sequence further illustrates our approach to handling ambiguous regions. As the first egg is broken, both the shells as the yolks are labeled with accurate object regions, maintaining the identity of the egg. It is, however, impossible to establish an accurate boundary between the transparent egg white and the bowl, so the annotators label it with a conservative ignore region. As more eggs are broken into the bowl, the yolks are still labeled with correct instance ids but it is impossible to separate the mixed egg whites, so the ignore label is maintained. In addition, this challenging video features fast motion due to objects gong out of frame, but our annotations correctly capture instance ids as the eggs re-appear.

1.2. Qualitative results

We now visualize the outputs of our AOT+ baseline on several sequences from Figure 6 in the main paper in the following video.

00:00-00:19 We begin with a success case where a peeled banana is accurately segmented as it is separated into several parts and its appearance changes. Notice that all the state transitions are smooth in this video and there is a strong contrast between the object and the background, making the task relatively easy for AOT+.

00:20-00:38 In the next sequence, however, although the appearance of the coffeemaker does not change as much, the transitions are more abrupt. Moreover, after the top part is separated and left on the metal sink it experiences full occlusion. This confuses our baseline, which still largely relies on appearance, resulting in the loss of that object part. Moreover, the model experiences several other small failures due to appearance similarity between the coffeemaker and the sink and reflections.

00:39-01:04 Over-reliance on appearance and limited spatio-temporal modeling capabilities of the model cause a complete failure in the next sequence, where two cuts of paper with nearly identical appearance are being rolled together. The model is able to distinguish the objects at first, while they are spatially separated, but as the two instances

Score	Definition
1	No visible object transformation. Either the verb was used in a different context or there was a mistake in the original annotation.
2	Technically there is a transformation in a video, but it only results in a negligible change of appearance and/or shape (e.g. folding a white towel in half or shaking a paint brush).
3	A noticeable transformation that nevertheless preserves the overall appearance and shape of the object (e.g. cutting an onion in half or opening the hood of a car).
4	A transformation that results in a significant change of the object shape and appearance (e.g. peeling a banana or breaking glass).
5	Complete change of object appearance, shape and texture (e.g breaking of an egg or grinding beans into flour).

Table 1. Definition of complexity scores used when filtering videos for VOST. These are by no means general, but they were helpful to formalize the process of video selection when constructing the dataset.

get mixed together and (self-)occluded it looses track of their identities.

01:04-01:36 We conclude with the egg breaking example. Here the model has to both deal with major appearance changes and distinguish between the two nearly identical instances. As the first egg is broken it only captures the shell in the beginning, failing to handle this challenging transformation. AOT+ manages to maintain the egg shells' identity at first, even though one of them goes out of frame, but ultimately fails at that too when the second egg is broken.

2. Additional details on Dataset Collection

In this section, we first provide the definitions of complexity categories that were used to select VOST videos and report the final distribution of complexity scores. We then report the instructions that were given to the annotators.

2.1. Complexity categories

Recall that, to focus on challenging object transformations, we labeled all videos from [4, 8] that contained a change of state verb [7, 10] in their original annotation with a complexity score on the scale from 1 to 5. In Table 1 we report the definitions of the scores that were used at this stage. Note that the problem of defining what constitutes a complex transformation is inherently ambiguous. The definitions we used are by no means general, but they were helpful to formalize the process of video selection when constructing VOST.

In Figure 1 we report the distribution of complexity score over all the labeled videos. Note that the total number of videos here is significantly lower that the raw number of clips extracted from Ego4D and EPIC-KITCHENS (10,706) because at this stage we also linked the clips representing a consecutive sequence of transformations (e.g. several cuts of the same onion) together. We can see that the wast majority of object transformations in the wild are not challenging, emphasizing the complexity of sourcing videos for VOST. That said, due to the large scale of [4, 8] we are still left with a sufficient number of videos in the target 4-5 range.

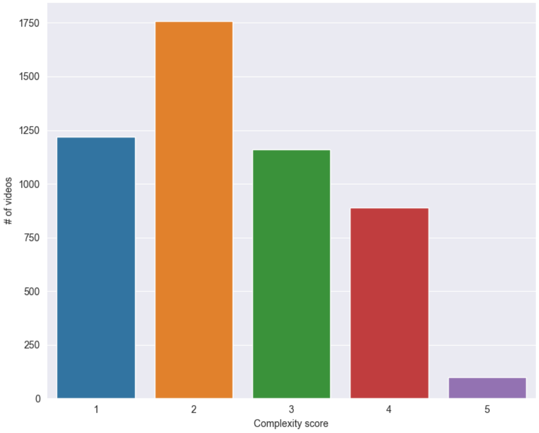

Figure 1. Distribution of complexity scores among reviewed clips. The majority of the transformations in the wild are not challenging but there is still a sufficient number of clips in the target 4-5 range.

2.2. Annotator instructions

We now report the instructions that were used by the annotators to label videos in VOST. The interface of the annotation tool is shown in Figure 4 in the main paper.

- The goal of this task is to provide polygon annotations for a wide variety of objects as they undergo transformations. The categories of the objects that need to be labeled in each video are provided in the "Label Categories" menu. If there are several objects of a certain category in a video, then only the ones that are being manipulated need to be labeled (e.g. if there are six eggs on a table but only two are broken, then only these 2 should be labeled).
- To label an object, select the appropriate category from the "Label Categories" menu, and then use the polygon tool to draw a polygon around it. A polygon is made up of a series of ordered points that you place around the object. The first and last points of the polygon must be the same and lines (edges) of a polygon cannot cross. When you place the first point, it will turn green. To

complete a polygon, close it by selecting the green start point again.

- There is a special label category "Ignore". It is only to be used in cases when an accurate polygon annotation is impossible to provide for a given region. In particular, there are 4 such scenarios:

  - Uncertain object boundaries due to motion blur. Label the non-blurred part with a regular polygon, and draw an "Ignore" polygon around the blurry one.
  - Tiny object parts that are too small to label to label accurately (e.g. tiny pieces of an onion skin). Draw the smallest possible "Ignore" polygon around each part.
  - (Semi)-transparent substances. Treat them in the same way as blurry boundaries (e.g. label the clearly visible part with a regular polygon, and only use the "Ignore" label for the uncertain region).
  - Parts of different objects that get mixed to the point at which they cannot be distinguished (e.g. two egg whites getting mixed together).

  The "Ignore" label should never be used in the first frame of a video.

- If an object is (partially) visible through another object (e.g. though a glass bottle), then the corresponding region should be labeled with the category of the front-most object. If that objects is not being labeled in the video, then the "Ignore" label should be used.

- Transformations can result in object splitting (such as breaking a glass). All the parts that results from splitting still need to be labeled (e.g. all the parts of the glass after it has been broken). This includes less obvious examples, such as a bowl wrapped in a plastic foil. As the bowl is getting unwrapped, both the bowl, its content and the plastic wrap need to be annotated. Another example which is worth noting is an egg. As it is getting cracked, both the resulting shells and the egg white/yolk need to be labeled.

- If there are multiple objects in a video, use the "Instance id" attribute to indicate which of the polygons belongs to which instance.

- Use the "Copy to next" icon to have the user interface copy all selected polygons (or all polygons if none are selected) in the current frame to the next frame. Use the "Copy to remaining frames" icon to copy all selected or all poylines to all subsequent frames.

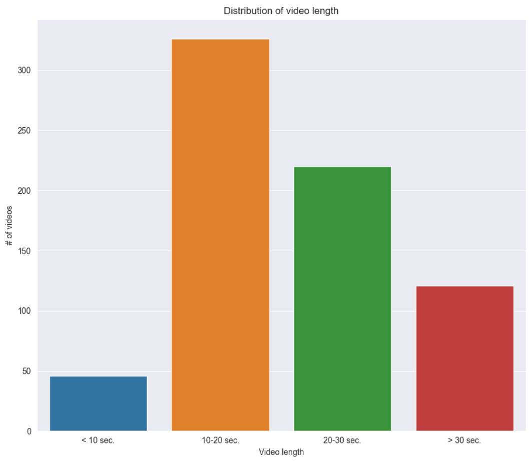

Figure 2. Distribution of video lengths in VOST. The vast majority of the samples fall into the challenging 10-30 seconds range, and a significant number of the videos are even longer than that.

- To adjust the location and shape of a polygon, select the polygon or the label associated with it from the "Labels" list in the menu on the right. Adjust the polygon by moving the points.

3. Dataset Statistics

In this section, we report additional statistics for the VOST dataset. We begin with Figure 2, which shows the distribution of clip lengths. The wast majority of the videos fall in the challenging 10-30 seconds range, which is significantly longer than in any existing VOS dataset. Moreover, 121 videos are longer than 30 seconds, capturing such lengthy transformations as grinding beans into flour.

Next, in Figure 3 we show the distribution over object mask sizes as a fraction of the whole image. Firstly, we can see that most objects in VOST are small, occupying less than 10% of the pixels in a frame. This is due to the nature of first-person videos, where the objects that are being manipulated are typically significantly smaller than the person who is manipulating them. That said, the distribution features a significant long tail of larger objects, such as cars or garbage bags, that can occupy more then half of the frame.

Finally, in Figure 4 we show the distribution of object motion at 5 fps, proportional to the size of the object. To this end, we follow [6] and compute the distance between the centers of bounding boxes enclosing the objects mask in frames t and $t-1$ in the horizontal dimension as $d_x^t = \frac{||x_{t-1}-x_t||}{a_{t-1}}$, where a_{t-1} is the bounding box area in frame $t-1$. The distance in the vertical dimension d_y^t is computed in the same way, and the combined distance is $d_t = ||d_x^t, d_y^t||_2^2$. We plot the largest motion in each video and observe that while most videos are temporally smooth

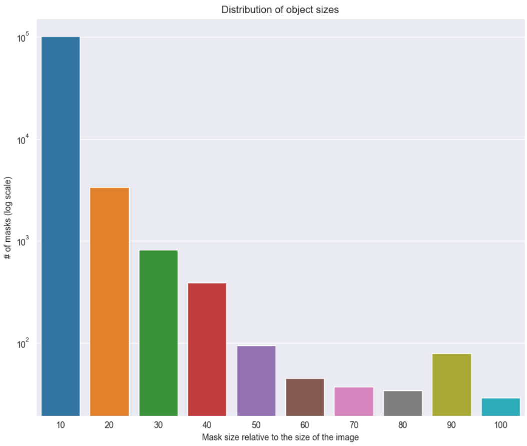

Figure 3. Distribution of object sizes in VOST. Most of the objects are small due to the nature of first-person videos, but there is a significant long tail of larger objects, such as cars.

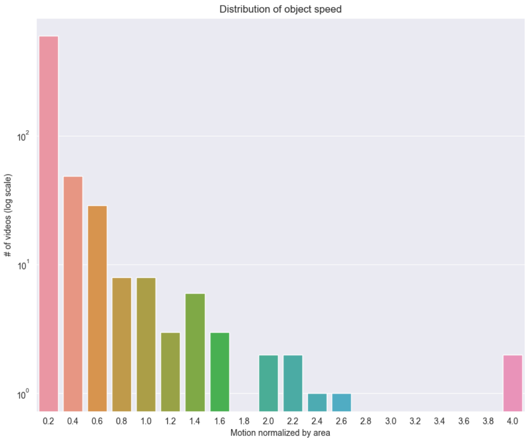

Figure 4. Distribution of object motion normalized by the object area in VOST. Most videos are smooth but there is a significant amount of challenging sequences with fast motion.

there is a significant number of clips with fast motion, which often correspond to the object going out of frame. As we saw in Table 3 in the main paper, such sequences are especially challenging for existing VOS algorithms.

4. Limitations of Annotation Interpolation

Several works have recently proposed to scale the size of VOS datasets by labeling at a very low fps and then interpolating ground truth labels to obtain temporally dense masks [5, 13]. As interpolation can fail, they automatically filter out the unreliable results and only keep the accurate trajectories. We now demonstrate that this approach fails precisely for the objects that undergo non-trivial transformations, justifying our decision to label VOST at 5 fps.

To this end, we visualize the interpolated labels from VISOR [5] for some of the sequences that feature object transformations in Figure 5. In the first video with onion peeling we can see that the interpolation fails as soon as the object starts to transform, with only the actor's hands accurately segmented. In the bag folding video in the top right interpolation succeeds in the middle of the sequence, but fails at the more challenging earlier and later parts. Note that the static boxes on the table, on the other hand, are perfectly segmented for the entire duration of the video. The cake cutting example in the bottom left of Figure 5 illustrates how interpolation fails to capture the part of the object that is separated from the rest of the cake. Finally, in the cheese cutting example in the bottom right interpolation fails for the entire duration of the sequence. Moreover, a part of the cheese is merged with the vegetables on the cutting board at the end. In contrast, VOST provides accurate, temporally dense labels even for the most challenging sequences.

5. Details of the R-STM architecture

We first provide a brief overview of the Long Short-Term Transformer (LSTT) architecture used in AOT [14], which we extend with a recurrent transformer module. We omit some of the unimportant details of LSTT architecture for brevity. Please see the original paper for a full description.

As the name suggests, LSTT combines two attention modules, $AttLT$ and $AttST$, that are implemented as transformers and are used to query long- and short-term memory respectively. Concretely,

$$AttLT(F^t, M^t) = Att(F^tW^Q, M^tW^K, M^tW^V), \quad (1)$$

where F^t is the feature encoding of the current frame, M^t is the memory state, Att is the standard, multi-head attention operation [12], and W^Q, W^K, W^V are linear projections. Crucially, the memory state M^t is simply a concatenation of per-frame feature maps from previous N timesteps: $M = Concat(F^{t_1}, F^{t_2}, ..., F^{t_N})$ combined with corresponding instance segmentation maps (either ground truth or predicted by the model). Short term memory is defined in the same way, with the main difference being that N is fixed to 1 in practice, so, effectively:

$$AttST = AttLT(F^t, F^{t-1}). \quad (2)$$

The outputs of both attention operations are then summed and the result is used to decode the instance masks of the target objects in the current frame.

It is easy to see that these attention operations perform appearance-based patch retrieval as the frame-level feature maps F can only encode, static appearance information. This is in stark contrast to traditional spatio-temporal memory modules [2, 11] that feature a single memory state tensor that is recurrently updated and can thus aggregate relevant information from the entire video. This is not only

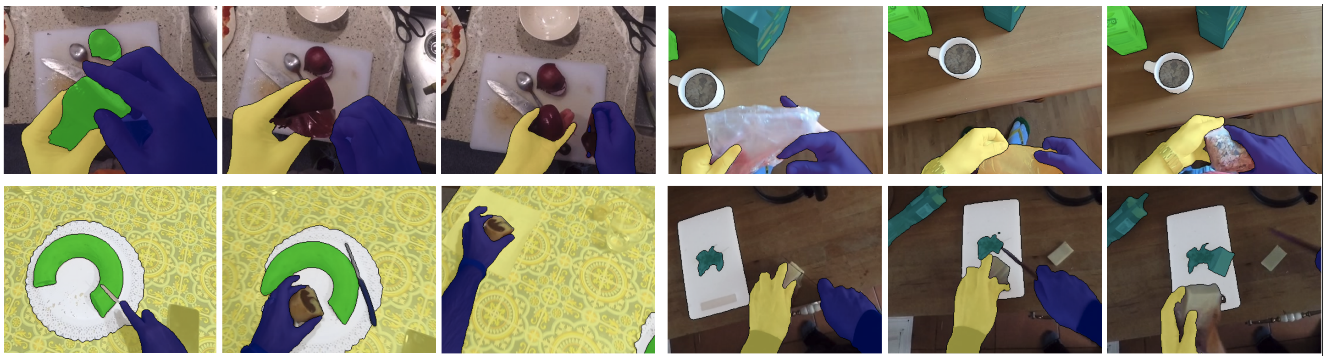

Figure 5. Visualization of automatically interpolated and filtered VISOR labels [5]. Colours indicate instance ids. We can see that automatic interpolation fails during transformations, such as peeling of the onion or folding of the cereal bag in the top row, whereas the objects with stable appearance, such as hands or boxes on the table are perfectly segmented. In VOST we are focusing precisely on the scenarios that automatic interpolation cannot handle, justifying our decision to densely label videos at 5 fps.

more computationally efficient than stacking feature maps, but also allows to represent concepts that are not explicitly present in any of the frames (e.g. locations of occluded objects).

To incorporate this capability into LSTT, we replace the short-term memory with a recurrent transformer ($R\text{-}STM$):

$$R\text{-}STM(F^t, M^t) = Att(F^t W^Q, K, V), \tag{3}$$

where $K = norm(W_F^K F^t + W_M^K M^t)$, similarly $V = norm(W_F^V F^t + W_M^V M^t)$, and $norm$ denotes layer normalization [1]. Crucially, $M^{t+1} = R\text{-}STM(F^t, M^t)$ making it a recurrent memory module. Our experiments in Table 4 in the main paper demonstrate that this simple modification indeed improves the transformation modeling capacity of AOT, but a more comprehensive approach for modeling spatio-temporal information is required to fully address the problem.

6. Further Implementation Details

All the models are trained and evaluated at 5 fps unless stated otherwise. When fine-tuning on VISOR [5] we excluded EPIC-KITCHENS [4] videos that were used in the validate or test sets of VOST. For AOT [14] and AOT+ we use the R50-L variant of the model and replace their default strategy of adding every fifth frame to the long term memory at inference time, which does not scale to long videos, with sparse insertion strategy proposed in [3]. We found that training CRW [9] at a higher 512×512 resolution leads to a slightly improved performance on VOST so we follow this strategy in our experiments. We also found that all baselines treat "Ignore" as another instance label. We modified their implementations to skip ignored regions in the first frame of a sequence and not include these pixels in the loss computation. Otherwise we left the original implementations and hyper-parameters unchanged for all of the methods, only adjusting the number of fine-tuning iterations on the validation set of VOST.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 4

[3] Ho Kei Cheng and Alexander G Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 5

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: collection, pipeline and challenges for EPCI-KITCHENS-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 2, 5

[5] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Ely Locke Higgins, Sanja Fidler, David Fouhey, and Dima Damen. EPIC-KITCHENS VISOR benchmark: Video segmentations and object relations. In *NeurIPS, Datasets and Benchmarks Track*, 2022. 1, 4, 5

[6] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. TAO: A large-scale benchmark for tracking any object. In *ECCV*, 2020. 3

[7] Charles J Fillmore. The grammar of hitting and breaking. 1967. 2

[8] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 2

[9] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 5

[10] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993. 2

[11] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM

network: A machine learning approach for precipitation nowcasting. *NeurIPS*, 2018. 4

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 4

[13] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 1, 4

[14] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 4, 5

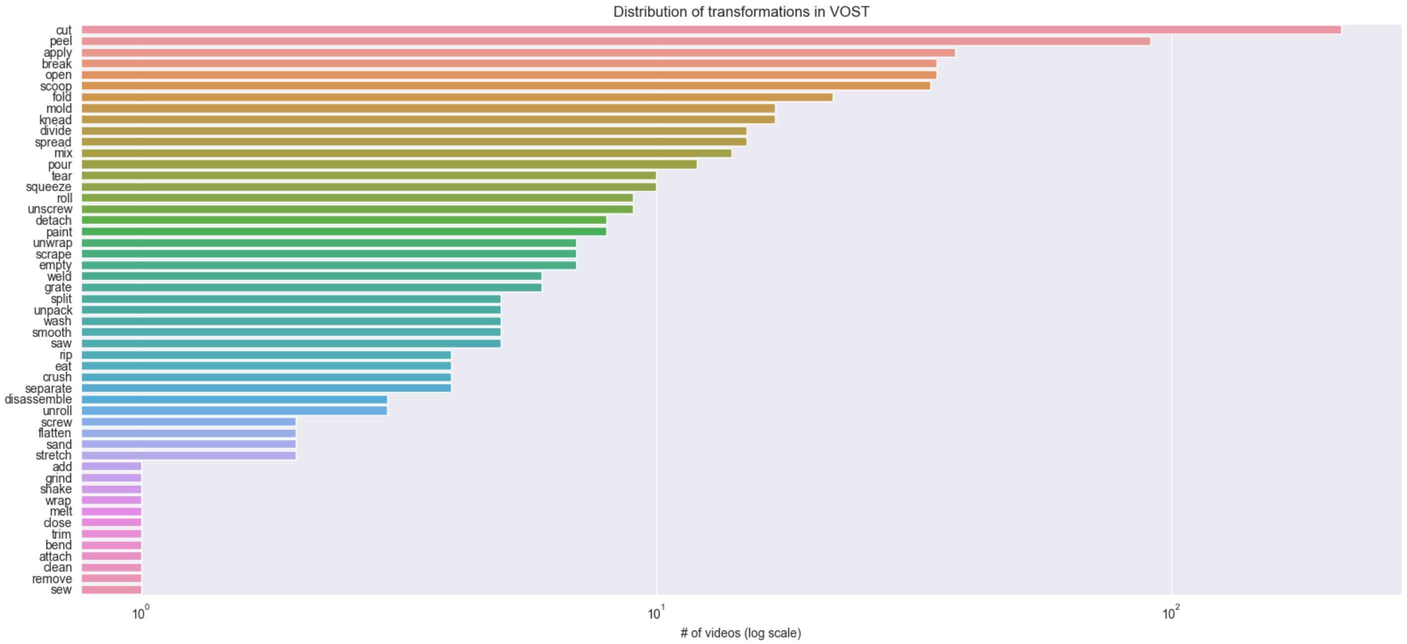

Figure 6. Distribution of transformations in VOST. While there is some bias towards common activities, like cutting and peeling, the tail of the distribution is sufficiently heavy.

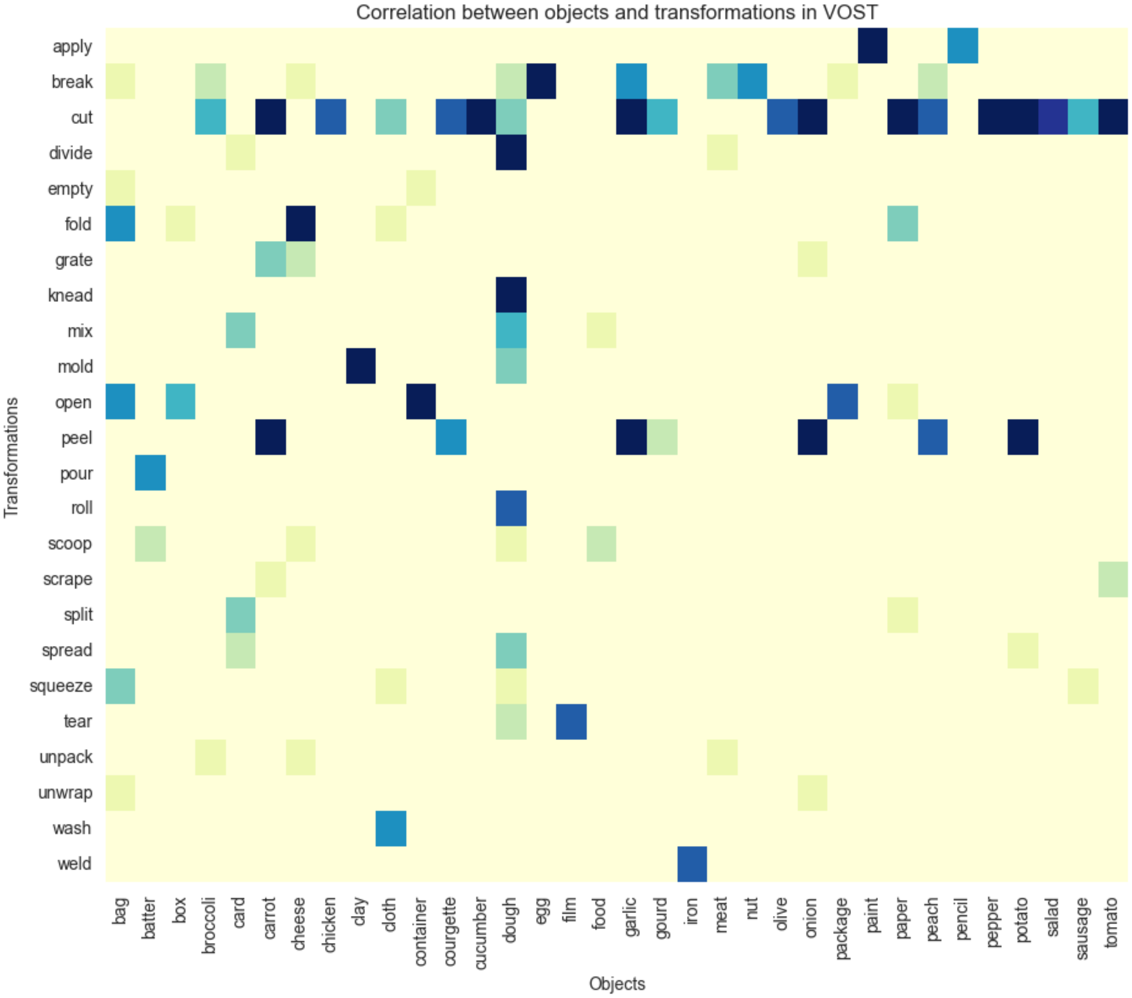

Figure 7. Co-occurrence statistics between the most common transformations and object categories in VOST. We observe that the most common transformation - cutting, has a very broad semantic meaning and can be applied to most objects. Overall, there is substantial entropy in the distribution, illustrating the diversity of VOST.