

SPOC: Spatially-Progressing Object State Change Segmentation in Video

Supplementary Material

The supplementary materials consist of:

- (A) A supplementary video offering a comprehensive overview of SPOC along with qualitative examples.
- (B) Detailed information on WhereToChange, elaborating on the data collection and annotation process. In addition, we analyze the dataset along various axes, highlighting its wide-ranging and expansive nature.
- (C) Experimental details: this includes complete experimental setup, baselines and metrics.
- (D) Extensive ablations of various component of SPOC with further results and visualizations that are referenced in the main paper.
- (E) Details of the downstream task on activity progress including metrics and baselines.
- (F) Discussion on limitations and scope for future work.

A. Video Containing Qualitative Results

We invite the reader to view the video available at <https://vision.cs.utexas.edu/projects/spoc-spatially-progressing-osc>, where we provide: (1) a comprehensive overview of SPOC, (2) video examples from WhereToChange, and (3) qualitative examples of SPOC’s predictions. These examples highlight SPOC’s ability in disambiguating object states across multiple object instances and effectively distinguishing actionable and transformed object regions. It delivers temporally smooth and coherent predictions that follow the natural OSC causal and temporal dynamics (from actionable to transformed), and shows strong performance even with novel OSCs not seen in training. Moreover, the activity progress curves highlight the importance of our proposed spatially-progressing task for downstream progress-monitoring and robotics applications. All these underscore the efficacy of SPOC, our proposed WhereToChange dataset, and our novel spatially-progressing task.

B. The WhereToChange Dataset

In this section, we describe in detail our annotation pipeline (Sec. B.2), analyze various properties of our WhereToChange dataset (Sec. B.3), and provide diverse samples from our dataset spanning a wide-range of state-changes and objects (Sec. B.4).

B.1. OSC Taxonomy

Our primary dataset, WTC-HowTo, is a large-scale collection derived from HowToChange [43], focusing on 10 state-change verbs that exhibit spatial progression behavior. These include actions such as chopping, coating, and

mashing, where object regions undergo sequential transformations from one state to another. From the 20 verbs in HowToChange, we select the 10 that demonstrate this characteristic, excluding verbs like blending, frying, and browning, which typically transform objects uniformly without intra-object state distinctions. The complete OSC object taxonomy is provided in Table A. Novel objects are absent for three verbs (coat, crush, melt) in WTC-HowTo, hence only seen objects are reported for these actions. Overall, our dataset encompasses a diverse range of objects across both subsets, providing a comprehensive benchmark for state-change understanding.

B.2. Annotation Pipeline for Eval Set

In this work, we present WhereToChange (WTC for short) as a large-scale video OSC dataset comprising fine-grained intra-object state-change segmentation. While the train split of WTC (~17k clips) relies on the pseudo-labeling procedure detailed in Sec. 3.2 & 3.3 for training SPOC (3.4), to ensure evaluation is rigorous and reliable, we collect manual annotations for 1162 eval clips from experienced professional human annotators.

Here, we describe the annotation process and guidelines followed for spatial-OSC annotations. For the 10 spatially-progressing verbs in HowToChange [43] (see Table 2), we first gather all the clips from the evaluation set of HowToChange, totalling 2787 clips. Then, we manually filter the clips, removing clips with high motion blur, no state changes, ambiguous actionable and transformed states, and so on. This leads to a final evaluation set of 1001 clips in WTC-HowTo. To curate the WTC-VOST evaluation set, we consider those verbs which overlap with WTC-HowTo (chop, peel). We gather clips from both train/val splits of VOST [40], while adopting a similar framework to filter out ambiguous and blurry clips. This yields 155 unique clips in WTC-VOST. In total, our evaluation set comprises 1162 clips across both subsets.

Next, with the help of 7 expert human annotators, the entire annotation process is conducted on TORAS [13] over the duration of a month, totaling 350 annotation hours. The annotators are first given a thorough outline of spatially-progressing OSCs and ways of annotating them, followed by multiple quizzes and sample sets to test their understanding. A screenshot of the annotation user interface is shown in Fig. A (i). An overview of the general guidelines provided to the annotators is shown in Fig. A (ii).

Given a video clip and the corresponding OSC (e.g. mashing potato), the annotators first review the clip, identifying the object in different states of change. Then they pro-

i) Overview of General Annotation Guidelines

Before beginning the annotations for a particular video and object state change (OSC), quickly skim through the video, to have a clear idea of how the two states (actionable and transformed) look like for that video.

Then proceed to annotate frame-by-frame. While annotating the frame, the following should be kept in mind:

- **Check if the frame is fit for annotation.** It should satisfy the following conditions:
 - The object of interest should be clearly visible and undergoing the specified state change.
 - Frame should not be too blurry (which means at least there is a tangible clarity on the objects present)
 - State change should not be too ambiguous i.e. you should be able to recognize the two states (actionable and transformed) in the frame
 - The object should not get intricately mixed up with other objects during the process of state change. E.g. For an OSC of crushing ginger, if the person adds other substances such as cloves, chillies, vegetables, etc into a mortar bowl and begins crushing, wherein the original object i.e. ginger is almost invisible, this frame should be marked as 'ignore'.
- **Carefully scan the frame to identify all instances of the object of interest.** We are interested in segmenting all instances, not just the active instance in hand. So identifying these instances in the first few frames will be crucial so that you can continue segmenting them in the rest of the frames.
- **Only annotate instances from the object of interest** - if there are other surrounding objects from other categories, make sure not to include those in the segmentation. E.g. For the OSC "grating carrot", you need to segment only carrots, but if the tray also has grated cheese, beets, etc, DO NOT segment those.
- **Exclude hands and tools** from segmentations - we are only interested in the object undergoing state change.
 - Please do not include entire tools. E.g. for grating, the grater should be excluded; for chopping, the knife should be excluded
 - If a tool is inevitably involved, try to exclude it as much as possible, unless it is intricately mixed up with the object covering it. E.g. For the OSC "mashing potato", if the tip of the fork used for mashing is in the mashed potato region and is covered with mashed potato, it is okay to include only the tip in the transformed mask.
- **Every frame need not always have actionable and transformed regions.** Some frames may also not contain both, in which case, you can move on to the next frame without any annotations. E.g. this can happen if the hands are covering the whole object
- **Segment all instances of the object in the image, not just the active object in hand.** It doesn't matter if the object is active or not--even if it is never interacted with, if it is of the same object type as the object of interest, then consider it. E.g. for the OSC "peeling banana", say a person is actively peeling a banana in hand, and there are some peeled and unpeeled bananas nearby which they never interact with. In addition to the active banana, all other instances are to be segmented under the relevant actionable or transformed categories.



Figure A. **Annotation User Interface and Guidelines.** (i) Overview of general annotation guidelines provided to the annotators. Verb-specific guidelines are in Table B. (ii) Annotation user interface on TORAS [13]. Details in Sec. B.2.

| Verb | Objects (seen) | Objects (novel) |
|------------------|---|--|
| WTC-HowTo | | |
| chopping | apple, avocado, bacon, banana, basil, broccoli, cabbage, carrot, celery, chicken, chili, chive, chocolate, cucumber, egg, garlic, ginger, jalapeno, leaf, lettuce, mango, mushroom, nut, onion, peanut, pecan, pepper, potato, scallion, shallot, strawberry, tomato, zucchini | almond, butter, capsicum, cauliflower, chilies, date, leek, pineapple, sausage |
| coating | apple, bread, cake | |
| crushing | biscuit, cooky, garlic, ginger, peanut, potato, strawberry, tomato | pepper |
| grating | apple, carrot, cheese, chocolate, cucumber, lemon, onion, orange, parmesan, potato, zucchini | butter, cauliflower, coconut, mozzarella |
| mashing | avocado, banana, chickpea, garlic, potato, strawberry, tomato | egg |
| melting | butter, candy, caramel, gelatin, ghee, jaggery, margarine, marshmallow, shortening, sugar | |
| mincing | beef, cilantro, garlic, ginger, jalapeno, meat, onion, parsley, scallion, shallot | carrot, pepper, tomato |
| peeling | apple, avocado, banana, beet, carrot, cucumber, egg, eggplant, garlic, ginger, lemon, mango, onion, orange, pear, plantain, potato, pumpkin, shrimp, squash, tomato, zucchini | kiwi, peach, pineapple, shallot |
| shredding | beef, cabbage, carrot, cheese, chicken, lettuce, meat, pork, potato, zucchini | coconut, mozzarella, parmesan |
| slicing | apple, avocado, bacon, banana, beef, bread, cabbage, cake, carrot, chicken, cucumber, egg, eggplant, ginger, leek, lemon, mango, meat, mushroom, onion, peach, pear, pepper, pineapple, potato, radish, sausage, scallion, shallot, steak, strawberry, tofu, tomato, watermelon, zucchini | butter, celery, jalapeno, lime, mozzarella, olive, orange, pepperoni |
| WTC-VOST | | |
| chopping | bacon, broccoli, carrot, celery, chicken, chili, cucumber, garlic, ginger, mango, onion, pepper, potato, scallion, spinach, tomato | asparagus, aubergine, beef, bread, butter, cake, corn, courgette, dough, gourd, ham, herbs, ladyfinger, mozzarella, pea, peach, pumpkin, salad |
| peeling | banana, carrot, garlic, onion, potato | aubergine, courgette, fish, gourd, peach, root |

Table A. **OSC taxonomy for WhereToChange**. WTC encompasses 116 objects undergoing 10 distinct state transitions, resulting in 232 unique OSCs (170 seen and 62 novel) across both WTC-HowTo and WTC-VOST.

ceed to annotate the video frame-by-frame, marking actionable and transformed regions of the relevant object. Frames where this distinction is unclear are marked as ignored frames. Ignored frames are not considered while computing the IoU metric during evaluation. Specific guidelines for each verb are listed in Table B. Once the annotations are complete, they are further vetted by us, and any edits are sent back to the annotators for corrections.

The aforementioned annotation procedure provides us with a high-quality evaluation set for WhereToChange, enabling us to conduct robust evaluation for the spatially-progressing OSC task. This data will be released publicly to allow further progress on the new task.

B.3. Dataset Analysis

We conduct a thorough analysis of our proposed WhereToChange dataset along different axes, highlighting its wide-ranging and diverse nature. A summary of the analysis is in Fig. B. We explain each of the properties in detail below.

Clip counts per verb We show the distribution of seen-novel object counts in WTC-HowTo (10 verbs) and WTC-Vost (3 verbs) in Fig. B (i). The OSC object taxonomy is in Table A. Novel objects are not present for 3 verbs in WTC-

HowTo (coat, crush, melt), hence only seen objects are reported. We see that our dataset comprises a wide collection of objects across both subsets. Our dataset allows for testing models for generalization across diverse object classes (e.g. while apple is a commonly occurring fruit in HowTo chopping videos, date is relatively low-frequency).

Clip duration per verb We show the duration of clips for every verb in Fig. B (ii). Clips are 25 seconds long on average across all verbs with a standard deviation of roughly 10 seconds. Our minimum clip duration is 3 seconds, while the maximum is 96 seconds long.

Duration of actionable and transformed phases Fig. B (iii) shows the duration of actionable and transformed stages for each verb. The actionable/transformed stage is those time-steps where an actionable/transformed mask is present in the annotation. We notice that some verbs have a large overlap between actionable and transformed stages (e.g. coat, peel), indicating a slower and longer transition period. Others see a smaller overlap (e.g. melt), indicating a quicker transition from actionable to transformed regions. This finding supports our intuitive understanding of the nature of these OSCs. Activities like chopping and grating are more drawn-out, involving human effort to act on the object and change it in slower stages. On the other hand, activities

| Verb | Guidelines |
|-----------|--|
| Chopping | <i>Actionable:</i> Whole fruits and large pieces <i>Transformed:</i> Only small pieces, slices, or rings Note: 1) If in a video, an object is cut in half and further chopped into smaller pieces, only the smaller pieces are transformed. The big halves initially are marked as actionable. 2) Hands or tools like knife, etc are excluded from the segmentation |
| Slicing | Same as above |
| Mincing | In addition to the above, mincing specifically means chopping into tiny pieces, not slices. So intermediate steps like slices are marked actionable |
| Grating | <i>Actionable:</i> Whole or chunked piece <i>Transformed:</i> Grated pieces Note: Grater and other tools are ignored |
| Shredding | Same as above |
| Mashing | <i>Actionable:</i> Whole fruit or large chunks <i>Transformed:</i> Mashed regions of fruit Note: The visual distinctions can be subtle, so textural difference are used as guidance. E.g. When mashing potatoes, unmashed regions are chunky, mashed regions are smooth |
| Crushing | Same as above |
| Peeling | <i>Actionable:</i> Unpeeled region <i>Transformed:</i> Peeled region Note: Peel that is detached from fruit is not segmented under either category. E.g. banana peel after being detached from fruit is not actionable or transformed, it is excluded from the segmentation |
| Melting | <i>Actionable:</i> Unmelted solid region <i>Transformed:</i> Melted liquid or paste-like region |
| Coating | <i>Actionable:</i> Object region plain and uncoated by external substance <i>Transformed:</i> Region coated with substance – e.g. bread coated with jam, nutella, butter, etc Note: a) Tools used for coating like spoon, etc are excluded as much as possible unless it is present on the object and prominently coated. b) Only the object of interest is segmented as transformed if coated, not the coating substance if it is stand-alone. E.g. containers containing the substance are not segmented as transformed. |

Table B. **Verb-specific annotation guidelines for WhereToChange.** General guidelines are in Fig. A (i).

like melting ghee or butter can proceed very quickly on account of automatic catalysts like heat, requiring little human time and effort.

Area of actionable and transformed regions Fig. B (iv) shows the area of actionable and transformed regions for each verb. We see that transformed regions occupy a larger area compared to actionable regions, likely indicating a larger surface area due to disintegration of the original object. For instance, chopping, slicing and grating disintegrate a whole fruit into multiple smaller pieces, considerably increasing surface area. This change is stark for mashing, where the volumetric object region is transformed into a flattened surface area. When comparing WTC-HowTo and WTC-VOST, the latter has overall smaller areas owing to the head-mounted camera, which captures a more zoomed

out view compared to close-up shots in the former.

Progression of actionable and transformed regions over time Another interesting aspect of our dataset is that due to the spatial segmentation annotation, we can directly track the behavior of the actionable and transformed regions over time. Unlike frame-level labels in [37, 43] or state-agnostic segmentations in [40, 46], our spatially-progressing OSC task has the unique benefit of fine-grained *spatial* and *temporal* state-change maps. Making use of this, we track the area of actionable and transformed regions over time and present averaged results across all clips per verb in Fig. B (v).

We observe that with time, the area of actionable regions decrease, approaching 0, while those of transformed regions increase, highlighting the natural progression of OSC dynamics present in our annotations. We also show the standard deviation in a lighter shade. Notice how the standard deviation for transformed regions tapers down to lower values with time, with the reverse being true for actionable. When comparing WTC-HowTo and WTC-VOST, we observe that the former often sees activities reaching completion (transformed area close to 0), while the latter sees higher values. This is because WTC-VOST comprises continuously captured real-time videos, where the activity seldom reaches completion within the short clip duration. However, the natural progression of shrinking actionable regions and expanding transformed regions is uniformly observed across both subsets and across all verbs.

B.4. Dataset Samples

We present representative samples from our WhereToChange evaluation set across all subsets (WTC-HowTo/WTC-VOST), data splits (Seen/Novel) and verbs in Figs. C, D, E. Fig. C shows a diverse collection of annotation samples for the 10 verbs containing spatial-OSCs in WTC-HowTo. Notice the fine-grained nature of the annotations, encapsulating precise boundaries of actionable and transformed regions (e.g. chopping chives, grating carrot, mashing tomato, and so on). In Fig. D, we depict frames within a clip sequence. We observe that with the passage of time, actionable regions progressively change into transformed regions, following the natural progression of the OSC. In Fig. E, we show frames and sequences from WTC-VOST, the out-of-distribution subset of WhereToChange. In this more challenging dataset comprising continuously captured egocentric videos, we yet again observe the natural OSC dynamics observed earlier viz smooth progression of actionable to transformed regions.

In summary, WhereToChange, is a wide-ranging dataset comprising spatially-progressing OSCs from a plethora of cooking activities. With fine-grained intra-object segmentations over a diverse set of seen and novel objects, we push the frontiers of video OSC understanding.

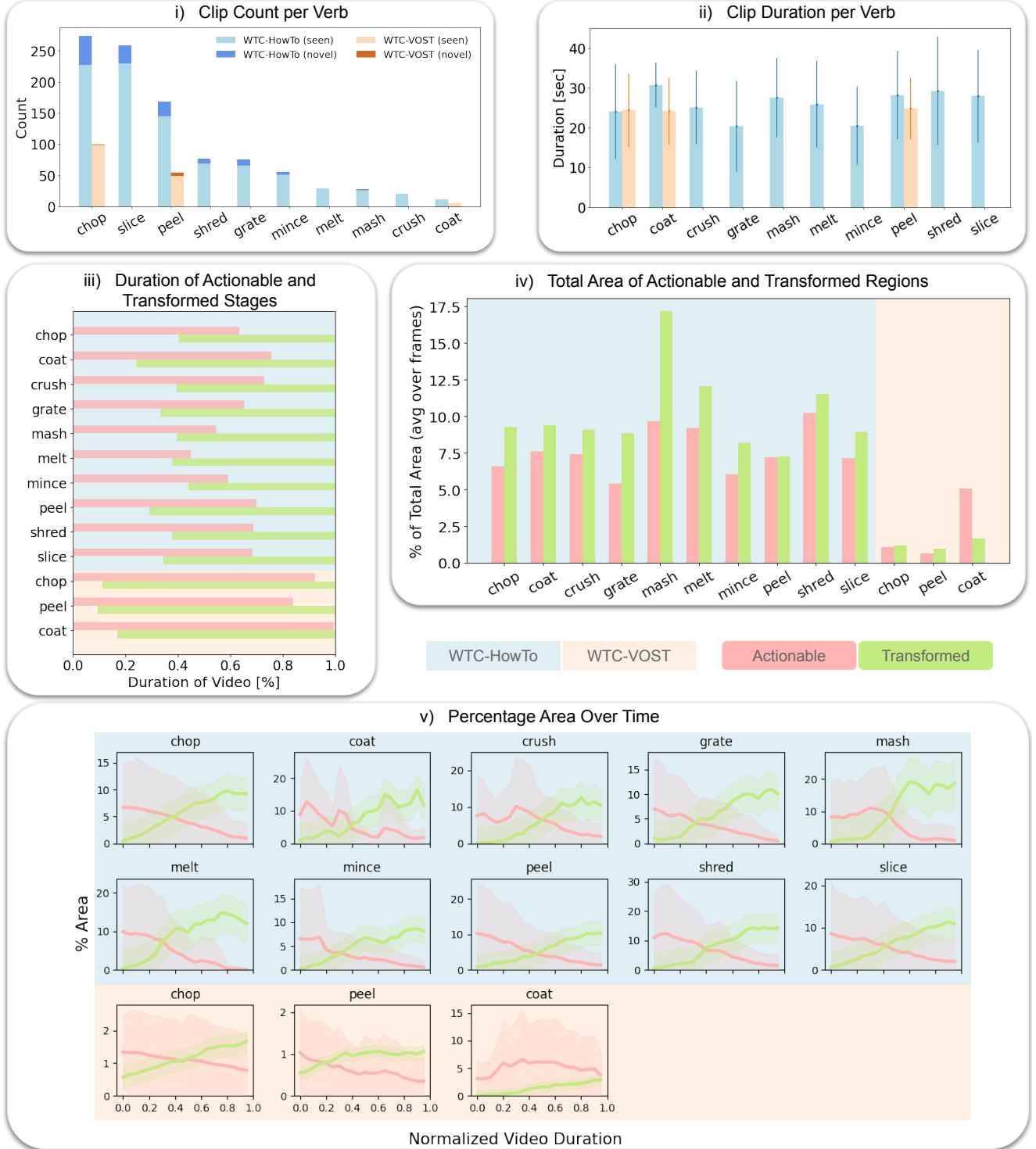


Figure B. **Analysis of WhereToChange dataset along different axes.** **i) Clip counts per verb:** distribution of seen-novel object counts in WTC-HowTo and WTC-Vost. Object taxonomy in Table A. **ii) Clip duration per verb:** clips are 20 seconds long on average across all verbs. **iii) Duration of actionable and transformed phases:** some verbs see a large overlap between actionable and transformed stages (e.g. coat, peel), indicating a slower and longer transition period. Others see a smaller overlap (e.g. melt), indicating a quicker transition from actionable to transformed regions. **iv) Area of actionable and transformed regions:** transformed regions occupy a larger area compared to actionable regions, likely indicating a larger surface area due to disintegration of the original object (e.g. chop, grate, mash). WTC-VOST has overall smaller areas compared to WTC-HowTo since the head-mounted camera captures a more zoomed out view compared to close-up shots in the latter. **v) Progression of actionable and transformed regions over time:** with time, the area of actionable regions decrease, while areas of transformed regions increase, highlighting the natural progression of OSC dynamics present in our annotations.



| WTC-HowTo: Diverse Frames | | | | | | | | | |
|---|---|---|--|--|---|---|--|--|--|
| Seen | | | | | Novel | | | | |
| Chop | | | | | | | | | |
| Mango | Chives | Strawberry | Chocolate | | Leek | Date | | | |
|  |  |  |  | |  |  | | | |
| Crush | | | | | | | | | |
| Mango | Cookie | Ginger | Potato | |  | | | | |
|  |  |  |  | | | | | | |
| Coat | | | | | | | | | |
| Cake | Apple | Bread | Apple | |  | | | | |
|  |  |  |  | | | | | | |
| Grate | | | | | | | | | |
| Cheese | Carrot | Orange | Zucchini | | Butter | Mozzarella | | | |
|  |  |  |  | |  |  | | | |
| Mash | | | | | | | | | |
| Banana | Tomato | Strawberry | Garlic | | Egg | Egg | | | |
|  |  |  |  | |  |  | | | |
| Melt | | | | | | | | | |
| Ghee | Margarine | Shortening | Butter | |  | | | | |
|  |  |  |  | | | | | | |
| Mince | | | | | | | | | |
| Onion | Shallot | Parsley | Jalapeno | | Tomato | Tomato | | | |
|  |  |  |  | |  |  | | | |
| Peel | | | | | | | | | |
| Avocado | Banana | Egg | Squash | | Peach | Pineapple | | | |
|  |  |  |  | |  |  | | | |
| Shred | | | | | | | | | |
| Lettuce | Pork | Potato | Meat | | Mozzarella | Mozzarella | | | |
|  |  |  |  | |  |  | | | |
| Slice | | | | | | | | | |
| Apple | Bread | Cake | Tofu | | Lime | Butter | | | |
|  |  |  |  | |  |  | | | |
| Actionable | | | | | Transformed | | | | |

Figure C. **Annotation samples from WhereToChange-HowTo.** Diverse frame samples from the HowToChange [43] subset of our proposed WhereToChange dataset. We show samples from both seen and novel object splits across all verbs and distinct nouns. Note that for three verbs (crush, coat, melt), we only have seen objects in the evaluation set, and hence omit novel object samples.



Figure D. **Annotation clip sequences from WhereToChange-HowTo.** We show sample frames from single clip sequences from the HowToChange [43] subset of our proposed WhereToChange dataset. Notice that with the passage of time, actionable regions progressively change into transformed regions, following the natural progression of the OSC.

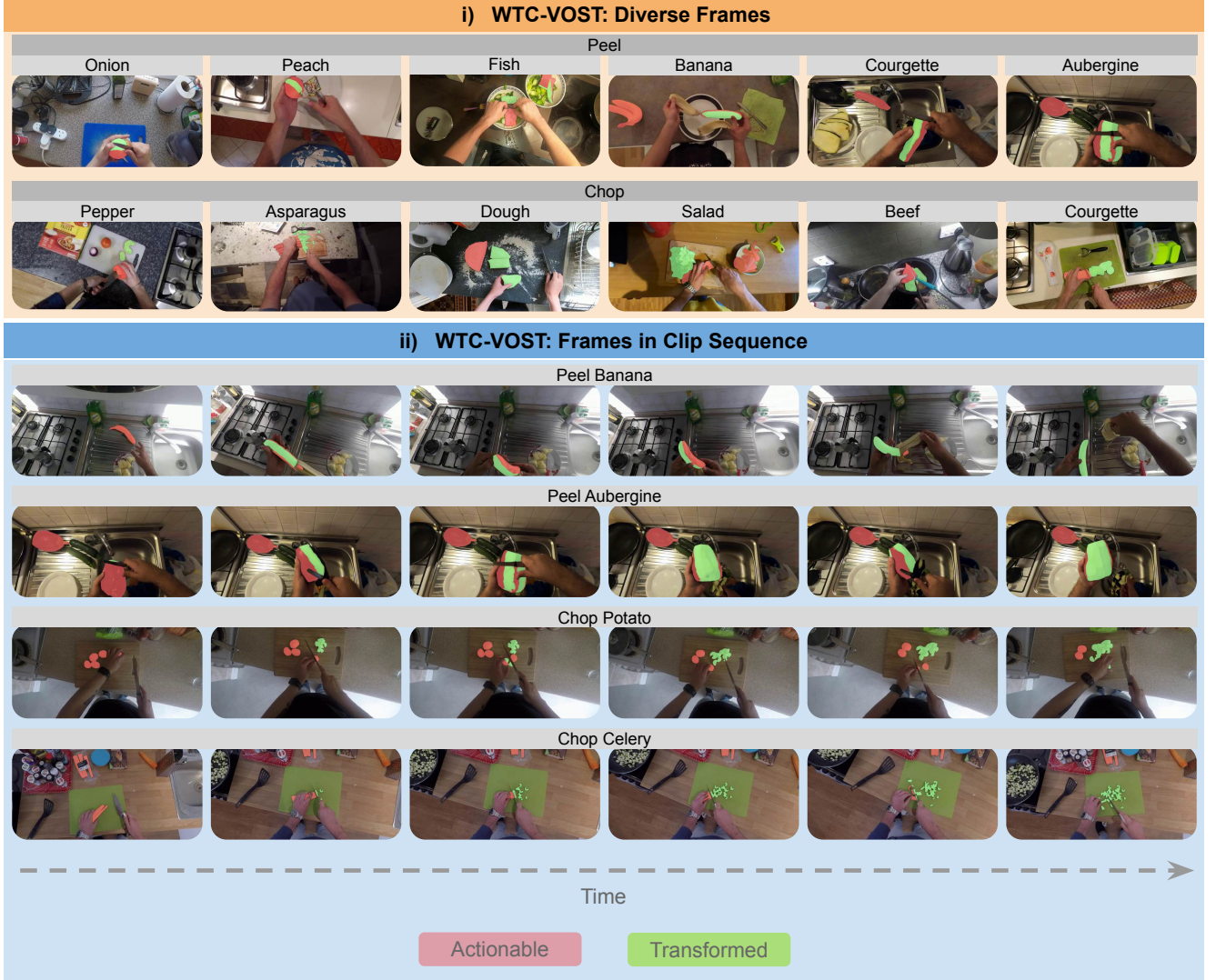


Figure E. **Annotation samples and clip sequences from WhereToChange-VOST.** We show sample frames (i) and clip sequences (ii) from the VOST [40] subset of our proposed WhereToChange dataset. This is a more challenging out-of-distribution dataset comprising continuously captured egocentric videos of human activities.

C. Experimental setup

In this section, we discuss in detail our experimental setup including implementation details, metrics and baselines.

C.1. Implementation details

Pseudo-labeling We list hyperparameter settings for each component of the pseudo-labeling stage of SPOC in Table C. We perform pseudo-labeling at 5 fps. In particular, we find that despite using a tracker [44], running the detector every at frequent intervals is necessary in order to re-identify dropped objects. Since objects morph rapidly with time, the tracker can often fail to track the objects meaningfully. Hence, we run the detector every 10 frames (2

seconds) in the video clip. For SAM, a 32×32 point grid is employed, and non-maximum suppression (NMS) with a threshold of 0.9 is applied to eliminate redundant mask proposals. In preliminary experiments on the choice of the CLIP vision encoder, we didn't find a significant performance difference between ViT-B/16 & ViT-B/32. We chose ViT-B/32 for faster pseudo-label generation at large scale. However, SPOC would naturally benefit from stronger representations; future use of more advanced backbones is a clear opportunity for further improvement.

Text Prompt For generating the text prompts for object detection, we automatically generate language labels with LLMs. We prompt ChatGPT-4 to generate actionable and transformed phrases for various OSCs following a few man-

ual examples (e.g. prompt: "OSC: chopping avocado, actionable: whole avocado, transformed: chopped avocado pieces. What is actionable and transformed label for OSC: slicing tomato?")

Model Training During training, our video model consists of a 3-layer transformer encoder (512 hidden dimension, 4 attention heads) and a 1-layer MLP decoder. While training the transformer model, we process 16 frames at a time sampled at 1 fps, w/ 16 global frame features and up to 4 mask features per frame, with both mask and time-positional embeddings. While the pseudo-labels include an "ambiguous" category, the SPOC model classifies each proposal into one of three states—actionable, transformed, or background. The ambiguous label is handled by preventing loss backpropagation for such proposals, ensuring they do not influence model training. We employ the AdamW optimizer with a learning rate of $1e-4$ and a weight decay of $1e-4$. Models are trained using a batch size of 64, over 50 epochs. Our inference speed is 4 FPS (on an Nvidia RTX 6000 GPU), with most time spent on mask proposal generation via SAM. Our model parameters are as follows: Trainable – SPOC Transformer (9.7M); Non-trainable – CLIP (88M), SAM (636M), DeAOT (49M); This is comparable to existing VOS models like DeAOT, and significantly faster than heavier models such as SAM (0.5 FPS, 636M params).

| Hyperparam | Value |
|---------------------|--------------|
| Grounding-Dino [18] | |
| box threshold | 0.35 |
| text threshold | 0.5 |
| box size threshold | 0.7 |
| SAM [35] | |
| model type | vit_h |
| prompt type | bounding box |
| Clip [33] | |
| model | ViT-B/32 |
| DeAOT [44] | |
| phase | PRE.YTB.DAV |
| model | r50_deaotl |
| long term mem gap | 9999 |
| max len long term | 9999 |
| SamTrack [7] | |
| sam gap | 10 |
| min area | 50 |
| max obj num | 255 |
| min new obj iou | 0.8 |

Table C. **Hyperparameter settings for different components during pseudo-labeling.** Details in Sec. C.

C.2. Baselines

We evaluate several state-of-the-art segmentation baselines for the task of spatially progressing object state change segmentation. These baselines fall into two broad categories:

pixel-based (MaskCLIP[50], MaskCLIP+[50]) and object-centric (GroundedSAM[35], SAMTrack[7], DEVA [5]).

- **Pixel-based** methods perform dense segmentation, classifying each pixel in the image into one of three regions: actionable, transformed, or background.
- **Object-centric** methods first employ off-the-shelf open-vocabulary detectors to identify relevant object regions before classifying them into the same three categories.

This structured evaluation allows us to compare the effectiveness of different approaches in capturing fine-grained state changes within objects.

We use official implementations of all baseline methods with their default hyperparameter setting values. Below, we explain each baseline in detail.

MaskCLIP [50] This is a state-of-the-art semantic segmentation method that leverages CLIP for pixel-level dense prediction to achieve annotation-free segmentation. To this end, keeping the pretrained CLIP weights frozen, they make minimal adaptations to generate pixel-level classification. Furthermore, MaskCLIP+ uses the output of MaskCLIP as pseudo-labels and trains a more advanced segmentation network. In the main paper Table 2, we report zero-shot results using MaskCLIP ViT-B/16. For the foreground prompt, given an OSC, e.g. 'chopping avocado', we prompt with 'chopped avocado pieces' and 'whole avocado'. In addition, following the original work, we also include background classes that are commonly present in the scene (e.g. person, hand, table, knife, etc) since it greatly improves the performance of MaskCLIP.

MaskCLIP+ [50] Further, we train MaskCLIP+, using an advanced pixel-level segmentation model (Deeplabv2-ResNet101) on the pseudo-labels generated by MaskCLIP on WTC-HowTo to obtain trained metrics. While training generally improves performance, we note that the rest of the object-centric methods (that use language-prompted detection followed by classification) still fare better compared to MaskCLIP+. This is because MaskCLIP still contains much noise in non-object regions owing to dense pixel-level predictions, while object-centric model predictions are mostly contained within relevant objects.

Grounded-SAM [35] We first use GroundingDino [18] to obtain text-prompted object bounding boxes. For e.g. for the OSC "chopping avocado", we use the prompts "whole avocado" and "chopped avocado pieces" to prompt GroundingDino to obtain actionable and transformed boxes. These boxes are then used to prompt SAM [14] to obtain masks. For the trained version, we use the pseudolabels generated by GroundedSAM to train our transformer model (Sec. 3.4). We find that the model fails to learn any reasonable information about states due to the poor pseudo-label quality that closely resembles random state assignment (compare with Random Label in Table 2).

SamTrack [7] In addition to intermittently detecting rele-

vant objects, this method also uses DeAOT [44], a tracker that tracks segmented object regions for the remainder of the video. We notice that large structural changes in the object can affect tracking, hence it becomes important to run the detector routinely to re-detect the dropped object and pass it back to the tracker.

Random Label In this baseline, we consider the object masks generated by SamTrack, however we assign “actionable” and “transformed” labels randomly throughout the video. A lot of the pseudo-labels generated by the baseline methods (e.g. GroundedSAM, SAMTrack, DEVA) all perform similarly to random label assignment. This underscores the challenging nature of our spatially-progressing OSC task—distinguishing between object states is a limiting factor of existing detection and segmentation methods.

DEVA [5] This is a decoupled video segmentation approach that uses XMem [4] to store and propagate masks. Similar to the pipeline in SAMTrack, an object detection [18] and segmentation [14] pipeline is adopted to first detect relevant object masks. These masks are later aggregated and propagated through the rest of the video using XMem. Akin to the above, we run routine detection (every 10 frames) to recapture dropped objects. The default setting in DEVA [5] generates a large number of mask proposals, drastically increasing run-time and GPU memory. For clips where we encounter memory overflows, we reduce max number of tracked objects from 100 to 20. We keep rest of the hyperparameters the same.

GroundedSAM [35] + GPT-4o [27] This part-centric baseline combines GroundedSAM for object segmentation with GPT-4o for intra-object classification. We sample a grid of points within each detected object mask and point-prompt SAM to generate candidate parts. These regions are then contoured and passed to GPT-4o with task-specific prompts to label each as {actionable, transformed, background}. For example, for peeling cucumber: “What does the contoured region correspond to? (a) unpeeled cucumber region, (b) peeled cucumber region, (c) not a cucumber.” See Fig. F. To maintain prompt reliability, we limit each image query to at most four contours, issuing multiple prompts if an image contains more parts. This baseline evaluates whether a state-of-the-art VLM can assign fine-grained, state-sensitive labels to object regions, beyond whole-object segmentation.

We find two main challenges on in-the-wild WTC videos: (a) Intra-object proposals are unstable—SAM2 often over-segments subtle texture changes or deformations, hallucinating “parts” where none exist. This is especially harmful for verbs like cutting, where the model may infer a slice despite no slicing occurring. (b) Labeling consistency over time is weak—the same region frequently flips between actionable and transformed, showing poor progress modeling. By contrast, SPOC’s causal ordering and tempo-

ral aggregation substantially reduce such errors.

We notice that object-centric methods outperform pixel-based methods in our task. They have the advantage of well-defined object regions provided by the object detection [18] and segmentation [14] pipeline, whereas pixel-based methods often have a greater degree of noise in the predictions. However, object-centric methods also suffer the disadvantage of being unable to assign two separate labels to regions within one detection. While the current SPOC model also follows an object-centric classification paradigm during training, the pseudo-labels could also be used to train dense pixel-based segmentation models. Future work could integrate object-centric and pixel-based techniques for optimal results.

C.3. Metrics

Following prior segmentation works [30, 35, 40, 46], we report mean IoU scores as a measure of segmentation performance. The reported mIoU is the average over actionable and transformed regions:

$$mIoU = \frac{mIoU_{act} + mIoU_{trf}}{2} \quad (2)$$

Following prior segmentation works [36, 40] that evaluate on video OSC datasets, we do not report boundary f-measure. This is because the objects undergoing OSC often change dramatically with constantly evolving boundaries, making it challenging to track and ascertain object boundaries.

D. SPOC Ablations

Effect of varying CLIP thresholds In Sec. 3.2, we make use of CLIP’s [33] vision-language embeddings to pseudo-label each mask proposal into one of (s_{act} , s_{trf} , s_{amb} , s_{bg}) based on similarity threshold values between the mask-vision and OSC-text embeddings. Eq. 1 follows a thresholding mechanism using $\Sigma s_{act, s_{trf}}$ and $\Delta s_{act, s_{trf}}$ values to set the pseudo-labels. We run a grid-search over Σ and Δ for each verb and compute IoU metrics on the generated pseudo-labels to choose the best threshold values. Fig. G shows a heatmap of $\Sigma(s_{act}, s_{trf})$ and $\Delta(s_{act}, s_{trf})$ and their respective IoU values. This heatmap shows aggregate values across all verbs. For our pseudo-labels, we choose best threshold values computed for each verb. This aggregates to a Σ value of 0.5 and a Δ value of 0.01.

Importance of regular detection We use an object detector and tracker to detect and track the relevant object for the duration of the state-change video. However, since the object often undergoes dramatic transformations such as shape, color or texture changes, the tracker might struggle to reliably track the object for prolonged periods. Further, while our task is a form of VOS, our formulation is designed to handle real-world instructional videos—where jump cuts

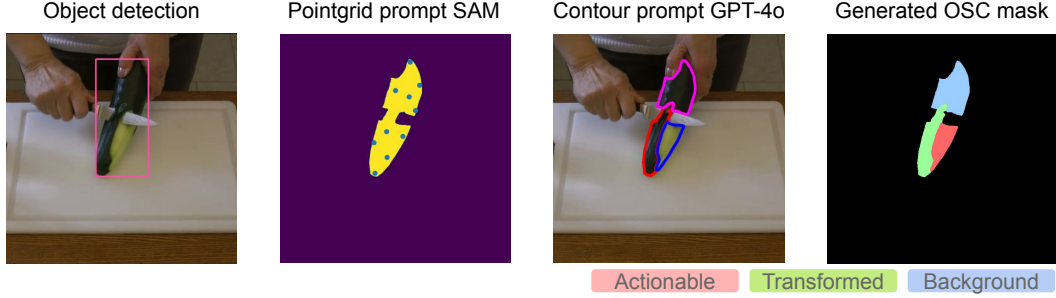


Figure F. **GPT-4o labeling pipeline.** We use [35] for object segmentation, extract intra-object parts via a point-grid prompt, and query GPT-4o to classify each part into one of actionable, transformed, or background. We notice two issues that arise: 1) parts are not reliably captured, often being hallucinated where none exist, or clubbing multiple distinct regions together 2) GPT-4o labels are less reliable, consistently flipping between the three labels throughout the video for the same segment.

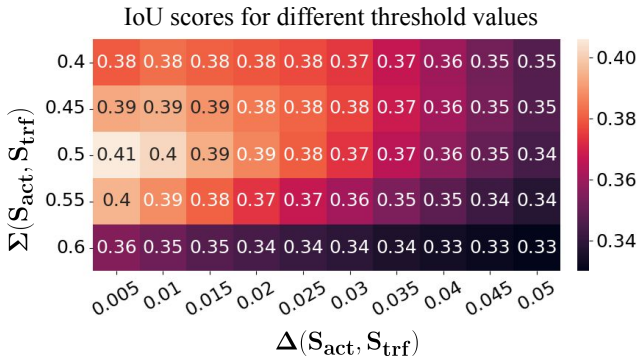


Figure G. **IoU scores for different CLIP similarity threshold values.** We plot a heatmap of $\Sigma(S_{act}, S_{trf})$ and $\Delta(S_{act}, S_{trf})$, CLIP similarity threshold values in Eq. 1. This heatmap shows aggregate values across all verbs. For our pseudo-labels, we choose best threshold values computed for each verb.

are common—and the object may move in and out of view. Rather than artificially require continuity (which would severely limit data diversity), we mitigate both the above issues by running the detector every 10 frames, passing potentially dropped objects back to the tracker. To ascertain the importance of regular detection, we run an ablation where we only detect the relevant object once at the start of the video while relying on the tracker’s predictions for the remainder of the frames. As seen in Table D, only first-frame detection and tracking yields 0.372 mIoU on WTC-HowTo—10% below SPOC (PL). Hence, regular detection is necessary to maintain object identity during tracking.

Importance of global-local representations As detailed in Sec. 3.4, SPOC is trained using a combination of global frame-level Dino-v2 [28] features and local mask-level CLIP [33] features as inputs. To ablate the importance of each of these features, we train two variants of SPOC (global-only and local-only) which only use the frame-level features and mask-level features respectively. In the global-

only setup, we pass bounding box locations as local inputs to differentiate between masks within the frame instead of CLIP features. We report results across 3 verbs in Table E. As seen, the combination of global and local features yields the best performance, highlighting the importance of both granularities for our task.

Upper-bound performance with oracle predictions We run an ablation to evaluate the upper-bound performance of the SPOC model with the existing object detector and mask tracking pipeline. Given mask proposals at each time-step, the trained SPOC model labels each into one of 3 categories: actionable, transformed or background. In our upper-bound experiment, we wish to gauge performance had SPOC made perfect predictions for each input proposal. To do this, we assign labels for each proposal from the ground truth annotations. If there is a large overlap of the proposal with GT_{act} , we assign s_{act} , for a large overlap with GT_{trf} , we assign s_{trf} , and s_{bg} if there is no significant overlap with either. This serves as an oracle upper-bound for SPOC given no changes in detection and tracking.

We report the mean IoU scores in Table F. The upper-bound oracle reaches an mIoU of 0.65, as against the SPOC trained model at 0.502. This indicates that while SPOC (trained) improves over pseudo-labels (0.455), further improvements in the training architecture could enhance SPOC performance even further. However, larger improvements in the off-the-shelf detector and tracker are necessary to push beyond the upper-bound score of 0.65 mIoU.

Additional qualitative results In addition to the results in Fig. 4, we provide additional qualitative results comparing SPOC with all baselines across all verbs from WTC-HowTo in Fig. H. We observe that SPOC predictions are more sensitive to object states, while the baselines are more prone to mis-identifying actionable and transformed object regions.

| Detection | WTC-HowTo | | | | | | | | | | |
|----------------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Mean | Chop | Crush | Coat | Grate | Mash | Melt | Mince | Peel | Shred | Slice |
| only 1st-frame detection | 0.372 | 0.405 | 0.332 | 0.184 | 0.389 | 0.488 | 0.344 | 0.450 | 0.348 | 0.385 | 0.396 |
| every 10th frame detection | 0.411 | 0.438 | 0.351 | 0.237 | 0.429 | 0.527 | 0.416 | 0.493 | 0.373 | 0.418 | 0.428 |

Table D. **Importance of regular detection while tracking.** To ascertain the importance of regular detection, we run an ablation where we only detect the relevant object once at the start of the video while relying on the tracker’s predictions for the remainder of the frames. As seen, this degrades performance since the objects can often dramatically change shape, color, or texture. Hence regular detection is necessary to maintain object identity during tracking.

| Model | Chop | Grate | Peel |
|---------------------|--------------|--------------|--------------|
| SPOC (global) | 0.423 | 0.469 | 0.401 |
| SPOC (local) | 0.504 | 0.508 | 0.432 |
| SPOC (global+local) | 0.523 | 0.528 | 0.449 |

Table E. **Importance of global and local features in SPOC model.** The combination of both frame-level global Dino features and mask-level local CLIP features yields the best performance.

E. Downstream Activity Progress

In Sec. 5, we proposed activity progress tracking as a downstream task to gauge the utility of spatially-progressing OSCs. We now provide details on the baselines and metrics.

E.1. Progress Baselines

Activity progress-monitoring is vital for AR/MR and robotics applications. In this regard, prior approaches [20, 21] to robot learning have sought to learn goal-based representations that can be recast as reward functions for training robots. We use these state-of-the-art approaches as baselines for our progress-monitoring task.

VIP Ma *et al.* [20] introduced VIP, a self-supervised visual reward and representation learning technique trained on Ego4D [11] for downstream robot tasks. Their key idea was to train an implicit value function that learns smooth and regular embeddings of egocentric video frames. From these frame-level embeddings, a goal-based reward function was formulated to be the goal-embedding distance difference at each time-step. Specifically, the reward at time-step t is defined as:

$$R(t) = \Phi(s_t) - \Phi(s_g) \quad (3)$$

where $\Phi(s_t)$ is the embedding at time-step t , and $\Phi(s_g)$ is the embedding at time-step T i.e. goal-image s_g . We refer the reader to [20] for further details. For our task, we supply the last image in the sequence as s_g . This yields goal-conditioned reward curves that double as progress curves.

LIV As a follow up to VIP, Ma *et al.* [21] introduced LIV as a dual language-image pretrained model that is capable of learning reward curves conditioned on both language and image embeddings. To adopt it for our task, we supply the

language goal to be relevant OSC (e.g. chopping carrot), and the image goal is the last frame in the clip sequence as defined earlier.

LIV-finetune Following the recommendations of the LIV paper, we finetune the pretrained LIV model on our WTC-HowTo training split and report results. We finetune for a total of 40 epochs and report results for the checkpoint yielding the best performance on the progress task.

E.2. Progress Metrics

We formulate progress metrics to evaluate two key properties that we would like to observe in the activity progress curves. First, since we consider irreversible state changes, the progress curves need to have a monotonic nature as the activity proceeds. Second, once the activity is complete, there should be no more changes in the curve values. This indicates the end of the task and should be denoted by stagnant curves. We now elaborate on each of the metrics.

Kendall’s Tau As stated above, activity progress curves need to reflect the monotonic nature of irreversible OSC progression. Here, we consider ideal curves to begin from a value of 1 (start of the activity, no progress) and finish at 0 (end of the activity, task completion). In other words, as the activity proceeds in time, the progress curve should smoothly proceed from 1 to 0, with a monotonically decreasing behavior for the duration of the activity. Kendall’s Tau is a statistical measure that can determine how well-aligned two sequences are in time. It has been used in prior video alignment works [9, 10] to measure temporal alignment in video frames.

For our task, we compute Kendall’s Tau (τ) of the curve as a measure of monotonicity. Since it evaluates the ordinal association between two quantities, it can be adapted to measure monotonicity in a single clip sequence as follows:

$$\tau = \frac{\text{number of increasing pairs} - \text{number of non-increasing pairs}}{\text{all possible pairs}} \quad (4)$$

where a datapoint in the curve is the progress value at a particular time-step in the sequence. A pair-wise metric compares pairs of time-steps throughout the sequence. In the equation above, a τ value of +1 indicates a perfectly monotonically increasing sequence, -1 indicates a perfectly monotonically decreasing sequence, and 0 indi-

| Model | WTC-HowTo | | | | | | | | | | |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | Mean | Chop | Crush | Coat | Grate | Mash | Melt | Mince | Peel | Shred | Slice |
| SPOC (pseudo-labels) | 0.455 | 0.461 | 0.383 | 0.447 | 0.447 | 0.566 | 0.418 | 0.500 | 0.417 | 0.453 | 0.458 |
| SPOC (trained) | 0.502 | 0.523 | 0.422 | 0.526 | 0.528 | 0.610 | 0.425 | 0.541 | 0.449 | 0.503 | 0.494 |
| SPOC (oracle) | 0.650 | 0.696 | 0.565 | 0.633 | 0.596 | 0.742 | 0.623 | 0.708 | 0.615 | 0.642 | 0.680 |

Table F. **Upper-bound oracle performance on WhereToChange-HowTo.** We assign labels to input mask proposals by comparing with ground truth annotation masks to obtain upper-bound oracle performance for SPOC. While the trained model improves over the pseudo-labels (including constraints), improvements in the training architecture could enhance SPOC performance even further. However, larger improvements in the off-the-shelf detector and tracker are necessary to push beyond the upper-bound score of 0.65 mIoU. Details in Sec. C.

cates no monotonicity. Hence, in our case, a high negative τ value is an indicator of good task progress.

In Table 4 of the main paper, OSC-based curves (SPOC-annotations and SPOC-model) have lower τ values compared to VIP [20] and LIV [21], the goal-based representation learning baselines. Contrary to the findings in the LIV paper regarding improved results with finetuning, we observe no significant difference in performance with or without finetuning. This underscores the challenging nature of our spatially-progressing OSC task. Approaches which merely rely on goal-images while placing no special emphasis on object-centric representations (such as object states in our case) can underperform for our task.

In addition to the curves in Fig. 5 in main, we show additional activity progress curves in Fig. I. We observe that OSC-based SPOC curves are generally more monotonic for the duration of the OSC, while the baselines tend to oscillate and remain stagnant for the most part.

End-state Sigma and L2 This metric evaluates curve behavior once the activity is complete. Ideally, once the end state is reached and the task is complete, we would not like to observe any fluctuations in the progress curve. To measure this, we compute both the variance of the curve (end_{σ}) and its absolute L2 value (end_{l_2}) in the end-state period. HowToChange has manual ground-truth annotations for classifying each frame in the clip into one of: initial, transitioning, end-state. We compute this metric in GT end-state annotated frames. Ideal values for both should be 0.

In Table 4 of the main paper, OSC-based curves have lower end_{σ} and end_{l_2} values compared to the baselines. This is corroborated by the qualitative figures in Figs. 5 and I, where the OSC-based curves stagnate in the end-state. In contrast, the goal-based baselines, being sensitive to the goal image, stagnate during the actual activity progression, while rapidly decreasing in the end-state after the task is complete. In certain scenarios where part of the object may be occluded (e.g. in Fig. I: mincing onion, hand occludes whole onion), OSC-based curves can show early saturation. The baselines nevertheless fail to track progress.

F. Limitations and Future Work

In addition to depicting failure cases in Fig. 4d of the main paper, here we discuss them in detail. We observe two main modes of failure in our model: single mask proposal for both actionable and transformed regions, and loss of object tracking. Samples from each failure mode are shown in Fig. J. We explain each case below.

The first bottleneck in our method is the detector we adopt to identify the object of interest. Existing open-world detectors have been trained to output bounding boxes for the whole object region. As a result, we find that when the actionable and transformed regions are very close to each other (e.g. Fig. J (i): chopping chive), the detector can output a single bounding box including both regions. As a result, we obtain a single label for both actionable and transformed regions, limiting the intra-object capability of our model (Table 3-Transition). This failure mode is less pronounced when there is a meaningful separation between the two regions (e.g. grating carrot, where the whole carrot and grated pieces are separated).

A second bottleneck for our method is the mask tracker we rely on for tracking detected masks in successive time-steps. Existing trackers work well for objects that remain static through time. Our dataset is primarily comprised of dynamically changing objects that often evolve drastically in terms of size, shape, texture, color and so on. This dynamic object morphing proves to be a challenge for existing trackers. As a result, we notice that tracked masks can sometimes taper out if the object is undergoing fast motions or transitions (e.g. Fig. J (ii): melting butter). As a result, running the detector at regular intervals becomes important, to offset some of the false negatives. The effect of this failure mode is more pronounced for WTC-VOST subset, which comprises continuously captured in-the-wild egocentric videos. Sudden, jerky head motions common in egocentric videos often throw off the tracker, leading to reduced segmentation performance.

Third, our baseline trains a separate model per verb, limiting generalization: unseen actions would require new training data. A promising future direction is to develop a multi-task verb-conditioned model, enabling transfer to unseen actions and objects with minimal supervision.

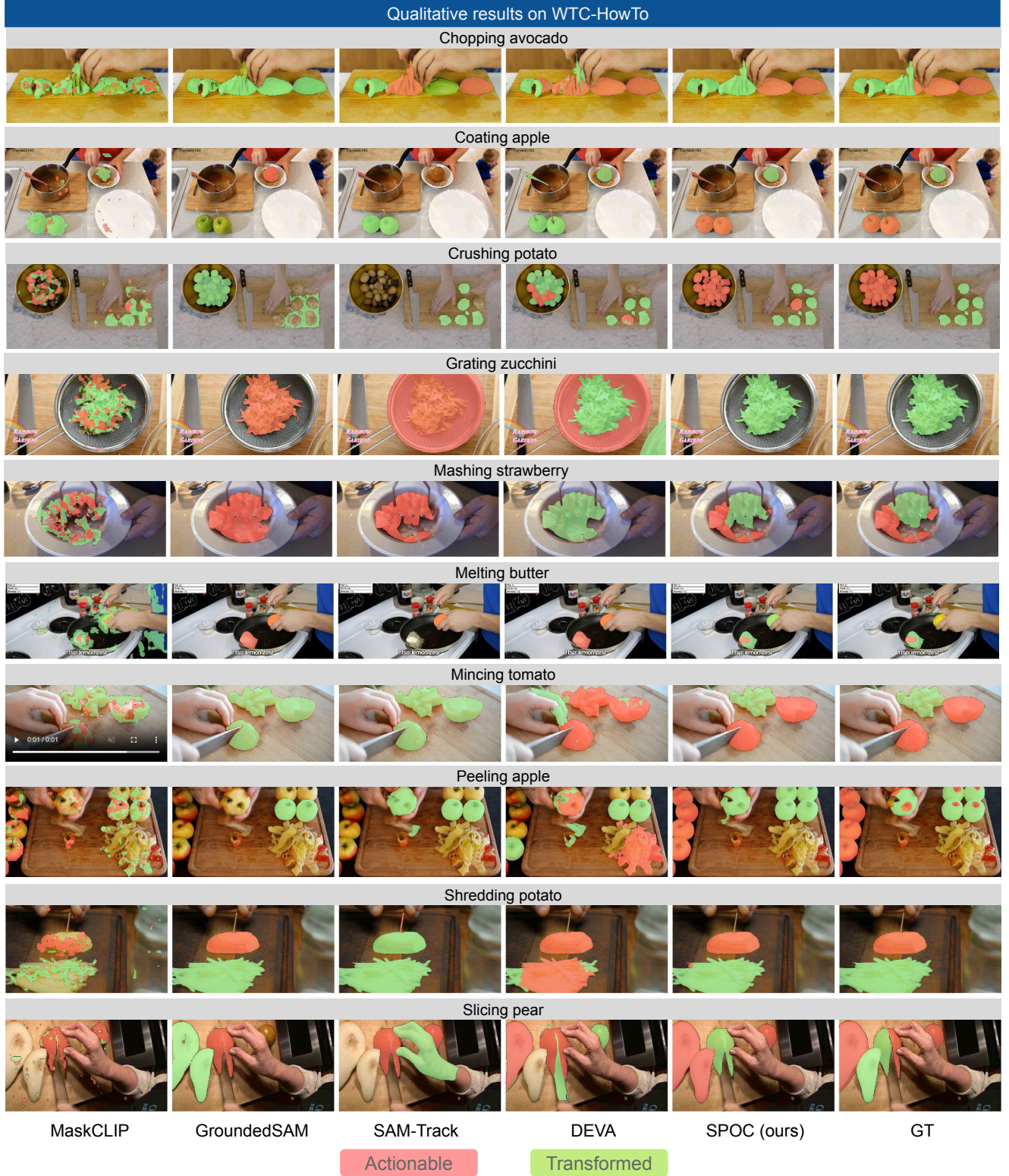


Figure H. **Qualitative results on WTC-HowTo subset.** In addition to the results in Fig. 4, we provide additional qualitative results comparing SPOC with all baselines across all verbs in WTC-HowTo. We observe that SPOC predictions are more sensitive to object states, while the baselines are more prone to mis-identifying actionable and transformed object regions.

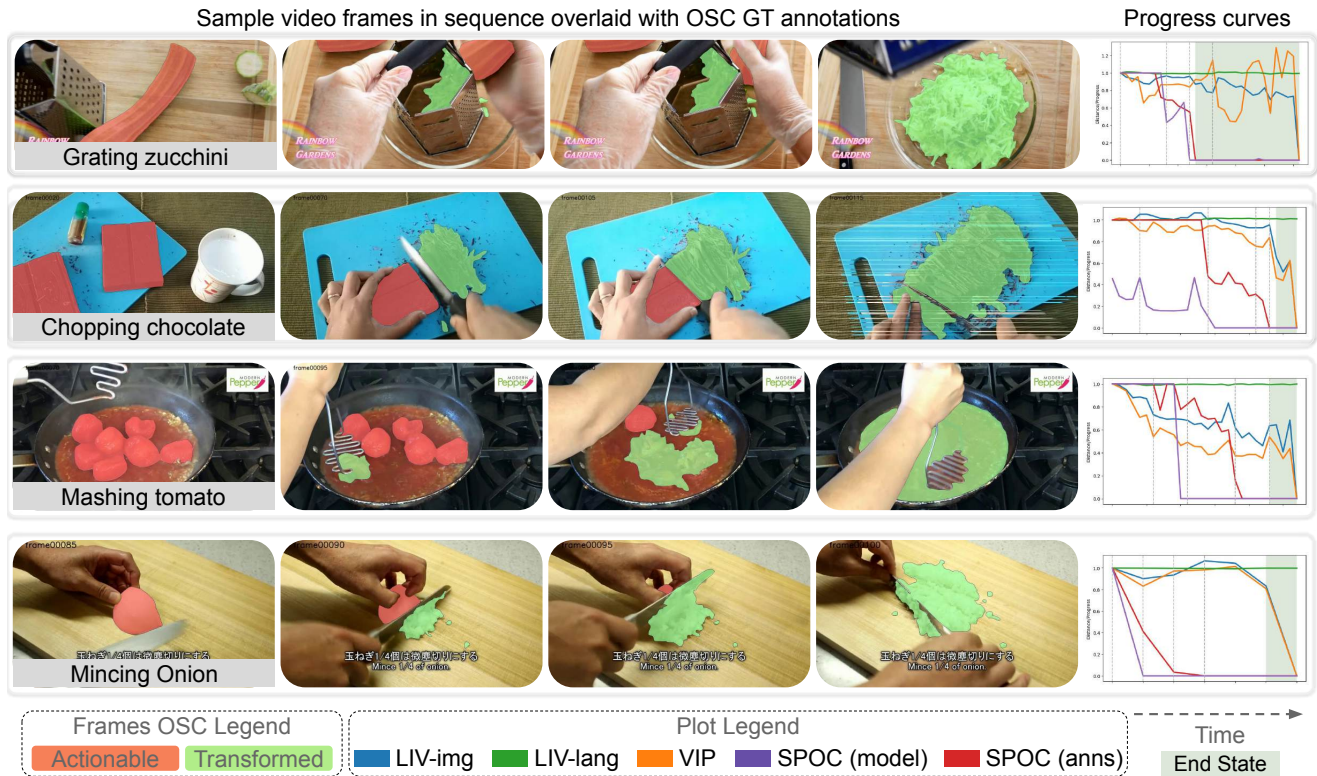


Figure I. **Activity progress curves.** Additional activity progress curves, like those in Fig. 5 in main. We show sample frames from a video sequence with progress curves generated by different methods, where vertical lines indicate the time-steps of sampled frames. Ideal curves should decrease monotonically, and saturate upon reaching the end state. In contrast to goal-based representation learning methods such as VIP [20] and LIV [21], OSC-based curves accurately track task progress, making them valuable for downstream applications like progress monitoring and robot learning.

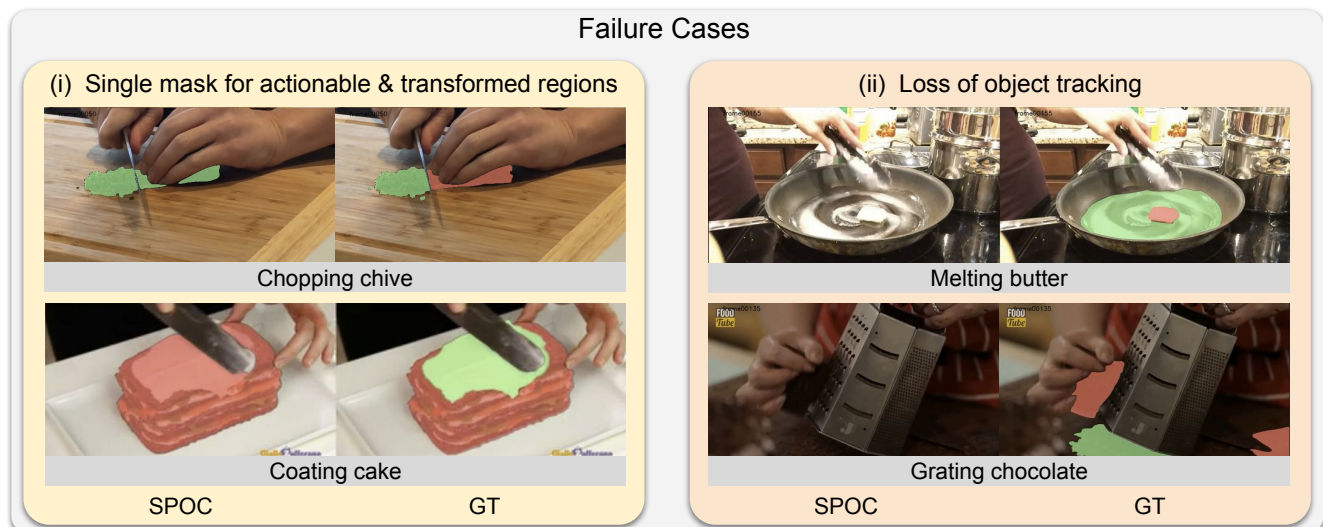


Figure J. **Failure modes.** We observe two main modes of failure in our model: (i) single mask proposal for both actionable and transformed regions (e.g. chopped and unchopped chive regions are within a single mask), and (ii) loss of object tracking (e.g. butter ceases to be tracked due to rapid motion). We discuss these cases in detail and discuss solutions in Sec. F.

Future advancements in object detection and mask tracking are likely to further enhance our model’s performance and alleviate the issues caused by these two failure modes. In addition, an exciting avenue to explore would be to predict mask proposals for actionable and transformed regions directly from the image. Such a method would have the dual benefit of learning intra-object mask proposal end-to-end while also being sensitive to OSC dynamics over time.