SPOC: Spatially-Progressing Object State Change Segmentation in Video

Priyanka Mandikal Tushar Nagarajan Alex Stoken Zihui Xue Kristen Grauman
The University of Texas at Austin
mandikal@utexas.edu

Abstract

Object state changes in video reveal critical information about human and agent activity. However, existing methods are limited to temporal localization of when the object is in its initial state (e.g., the unchopped avocado) versus when it has completed a state change (e.g., the chopped avocado), which limits applicability for any task requiring detailed information about the progress of the actions and its spatial localization. We propose to deepen the problem by introducing the spatially-progressing object state change segmentation task. The goal is to segment at the pixel-level those regions of an object that are actionable and those that are transformed. We introduce the first model to address this task, designing a VLM-based pseudo-labeling approach, state-change dynamics constraints, and a novel WhereToChange benchmark built on in-the-wild Internet videos. Experiments on two datasets validate both the challenge of the new task as well as the promise of our model for localizing exactly where and how fast objects are changing in video. We further demonstrate useful implications for tracking activity progress to benefit robotic agents.

1. Introduction

Understanding object state changes (OSC) is crucial for various applications in computer vision and robotics. OSC refers to the dynamic process where objects transition from one state to another, such as a whole apple being sliced, a potato being mashed, or a piece of cloth being ironed. Accurately capturing and analyzing these transitions is essential for tasks that require real-time decision-making and action planning. For instance, in human activity videos, detecting state changes could enable AR/MR assistants to monitor the progress of activities and prompt timely instructions and interventions. In robotics, precise OSC understanding would enable robots to perform complex tasks, such as cooking or assembly, by determining when and where to execute the next action.

Project webpage: https://vision.cs.utexas.edu/projects/spoc-spatially-progressing-osc

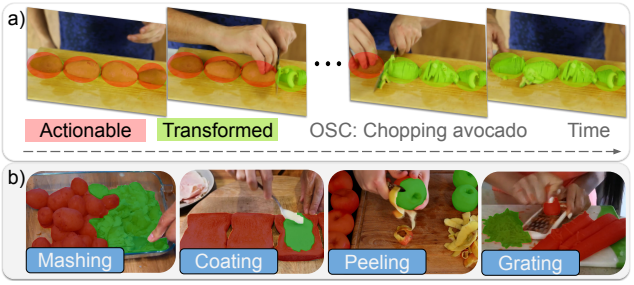

Figure 1. a) An illustration of the spatially-progressing video OSC segmentation problem: With time, the regions within an object undergo a progressive state-change from actionable to transformed. b) A diverse set of spatially-progressing segmentations for different state-change activities. Red is actionable; green is transformed.

Current approaches for OSC typically classify video frames as initial, transitioning, or end states [36, 37, 42] or segment entire objects undergoing OSCs [39, 45]. However, existing work makes an important assumption that all parts of the interacted objects change simultaneously and uniformly over time. We observe that in reality, an OSC often also *spatially progresses during an action*: (1) state changes occur to specific localized object parts or regions over time, and (2) parts of objects can undergo changes non-uniformly. For example, in Figure 1, the avocado is chopped from right to left, and parts of the potato mixture get mashed at different rates. Though not possible with existing methods, these pixel-level spatial distinctions are essential for planning and execution of any such state-changing process—such as a robot deciding exactly *where* to act next, or an AI assistant capturing a smart-glasses user's progress on their task.

To address this fundamental limitation, we introduce a novel task formulation: spatially-progressing OSC segmentation (SPOC). Given an OSC, we formally categorize the object regions into one of two states—*actionable* or *transformed*. For example, in Fig. 1a, the chopped region of the avocado is transformed, while the yet-to-be-chopped region is actionable. Given a video, the goal is to actively segment and label these regions in each frame as the object undergoes the state change transition. Note that not all OSCs are spatially-progressing. We define a spatially-progressing OSC as a state change that sequentially affects different re-

gions of the object. This includes a wide variety of human activities like chopping, peeling, cleaning, painting and so on, and excludes those where the whole object undergoes state change uniformly, like grilling, frying, or blending[1].

Our approach copes with the intrinsic challenges of spatial OSC, where objects can undergo dramatic transformations in color, texture, shape, topology, and size, often bearing little resemblance to their original form. These challenges thwart direct application of today's video object segmentation models [34, 49], as we will see in results, and call for new training resources and objectives to be developed.

Equipped with this new task formulation, we propose two key innovations to tackle the spatially-progressing OSC problem. First, we design a way to acquire a large-scale training dataset by pseudo-labeling human activity videos using vision-language models (VLMs). We show that by carefully combining segmentation models with language-image embeddings, we can generate a weakly-labeled pre-training dataset sufficient to bootstrap the training process. Second, we introduce novel state-change dynamics constraints that encourage the pseudo-labels to satisfy two key criteria defining state change dynamics: causal ordering and ambiguity resolution. These constraints are particularly effective in correcting errors where the VLM may have misclassified ambiguous regions. Using both ideas, we then train a video model to label mask proposals for distinct object states across relevant objects within each video frame.

Complementing our task and model contributions, we present WhereToChange, the first benchmark for spatially-progressing video OSC, with fine-grained intra-object state-change segmentations (Table 1). Derived from HowToChange [42] and VOST [39], our dataset includes human-annotated OSC segmentations for 1162 human-activity video clips, spanning 210 state changes and 102 unique objects. Experiments on WhereToChange demonstrate that our model significantly outperforms state-of-the-art methods adapted to address this novel problem. Furthermore, we show how to deploy our model to enhance agent learning, using its fine-grained object state understanding to track activity progress over time—showing promising results that can drive reward functions for robot manipulation. Overall, our work is an important step toward finer understanding of object state changes in videos.

2. Related Work

Object state changes Current approaches to OSC understanding fall into three main areas: classification [36, 37, 42], temporal segmentation [39, 45], and generation [38]. Early OSC methods focused on image-level classification [12, 15, 22, 24–26, 30, 31], later expanding to video-based approaches that leverage the dynamics of OSCs [1, 19, 36]. The HowToChange dataset [42] further extended classification to unseen object categories. Recently, VOST [39] and VSCOS [46] introduced the task of segmenting objects undergoing state changes in videos, but only address the segmentation of the *entire* object throughout the video. In contrast, we provide spatial annotations of state changes both *within* and *across* object instances, for a more detailed benchmark for video object segmentation. Additionally, our OSC dynamics operate at the individual object level rather than at a broader frame level.

Video object segmentation Video object segmentation (VOS) involves the pixel-level separation of foreground objects from the background in videos. Extensive research addresses semi-supervised VOS [4, 43, 44], which segments target objects throughout the video based on segmentation masks manually provided in the initial frames. Popular VOS datasets like DAVIS [2, 29] and YouTube-VOS [41] focus on objects that maintain their structure and integrity throughout the video, such as people, trees, and cars. More recent VOS resources incorporate state-agnostic segmentation of state-changing objects, but at the whole-object level [39, 45]. Our formulation and benchmark go one significant step further, requiring segmentation of the (intra-object) spatio-temporal state change progress.

Open-vocabulary segmentation Segment-Anything (SAM) [14, 33] is a promptable segmentation model with zero-shot generalization to unfamiliar objects and images. While SAM originally worked with click or bounding-box prompts, several subsequent works integrate SAM with textual prompts in an open-vocabulary setting [5, 7, 34, 35], to segment all object instances belonging to the categories described. These methods fall short for our task since they are often unable to distinguish between different states of the object, as we show in results. However, our strategy to generate large-scale pseudo-labels from VLMs [32] adhering to specific state-change constraints can infuse this object state-sensitive information into model training, yielding significantly stronger performance.

Vision and language Numerous studies have leveraged the robust vision-language embeddings of CLIP [32] for various downstream applications [17, 28, 40, 47]. However, research has highlighted the limitations of using CLIP directly for tasks like generating bounding boxes or segmentation masks, as it was initially trained on image-text pairs and lacks pixel-level detail [16, 47–49]. Recently, ROSA [35] adapted CLIP to represent local regions and phrases for object segmentation in narrational videos. VidOSC [42] utilizes CLIP for generating pseudo-labels to build the HowToChange dataset, but it only generates temporal labels at a global frame level. Our approach and motivation are distinct in two keys ways. First, we focus on *intra-object* segmentation of state change transitions, rather than segment-

[1] We consider cooking-related OSCs due to their real-world value, complexity, wide diversity, and data availability; however our approach is generalizable to other domains.

ing the entire object as a single instance like ROSA [35], or predicting global frame-level labels like VidOSC [42]. Second, we introduce novel object-level state-change dynamics constraints that play an influential role for our task.

3. Approach

We present SPOC, a framework for spatially-progressing OSC segmentation. We present the problem formulation (Sec. 3.1), followed by our approach to generate pseudo-labels to train our model (Sec. 3.2) and incorporate OSC dynamics constraints (Sec. 3.3). Finally, we present our model-design and training details for state-change sensitive object segmentation (Sec. 3.4).

3.1. The Spatially-Progressing OSC Task

We cast the problem as a state-change dependent video segmentation task. More formally, given a sequence of video frames $\{I_1, I_2, \ldots, I_T\}$, where I_t represents the video frame at time-step t, the objective is to generate segmentation masks $\{L_t^0, L_t^1, \ldots, L_t^K\}$ for each frame I_t in the video sequence. Each mask L_t^k should accurately represent the spatial regions of the k-th object region in one of two states (*actionable* or *transformed*)[2] at time-step t. The states are defined as follows: the actionable state (s_{act}) is the state of the object region before any change has taken place (e.g., whole avocado), and the transformed state (s_{trf}) is the state of the object region after transformation (e.g., chopped avocado piece). We say that all regions that do not yet adhere to the final state are actionable (e.g. intermediate states before fully chopped). This reduces ambiguity and allows a clear binary formulation for our task. Note that one object can have multiple regions simultaneously in different states. We focus specifically on cooking activity videos, which include a wide variety of complex and common OSCs, supported by available datasets [8, 11, 23] (see Table 2 for the full set).

3.2. Pseudo-Label Generation

Existing state-of-the-art open-set object detection methods [6, 18] work well for broad object categories. However, identifying objects with specific fine-grained state attributes—especially those that drastically alter the object's appearance—remains challenging. To address this, we train a model specifically for state-change disambiguation, first leveraging VLMs in a novel approach to generate large-scale pseudo-labeled data by employing a pseudo-label generation process (Fig. 2a) as follows:

Mask Proposal Generation Given a video frame I_t, we detect and localize objects by name (e.g., "avocado") using an open-vocabulary object detector [34] to produce bounding boxes $B_t^{0..K}$. We assume that each video clip is associated with a single OSC, which is typical in in-the-wild videos [8, 11, 42]. Using these bounding boxes, we generate segmentation masks $M_t^{0..K}$ for each detected object, leveraging a pre-trained segmentation model [34] to produce high-quality masks. These masks are then tracked [43] across frames to maintain consistency, capturing object state transitions over time. This tracking also enables the application of OSC dynamics constraints, as defined below.

Mask Labeling To distinguish between states s_{act} and s_{trf}, we integrate the CLIP [32] model. CLIP's vision-language embeddings are utilized to generate semantic labels through a thresholding process that aligns visual features of object regions with descriptive text annotations of the state change. To label a mask, we first obtain a vision embedding z_v for the object region within the mask. Then, we compute textual embeddings for the two state descriptions (e.g., t_{act}="whole avocado", t_{trf}="chopped avocado pieces") to produce $z_t \in \mathbb{R}^{d\times 2}$, where d is the embedding dimension. These descriptions are automatically generated by few-shot prompting an LLM with the OSC verb and noun. We then compute similarity scores between the vision-language embeddings by taking a simple dot product $\mathbf{S} = z_v \cdot z_t$, where $\mathbf{S} \in \mathbb{R}^2$ is a similarity score matrix. We can then obtain the pseudo-label for the object region as follows:

$$\hat{y}_t = \begin{cases} \text{Background} & \text{if } \mathbf{S}^t_{act} + \mathbf{S}^t_{trf} < \tau \\ \text{Ambiguous} & \text{elif } |\mathbf{S}^t_{act} - \mathbf{S}^t_{trf}| < \delta \\ \text{Actionable} & \text{elif } \mathbf{S}^t_{act} > \mathbf{S}^t_{trf} \\ \text{Transformed} & \text{elif } \mathbf{S}^t_{trf} > \mathbf{S}^t_{act}. \end{cases} \quad (1)$$

Here, δ is the threshold that separates states, and τ is the threshold that differentiates between states and background. Similar to [42], if the VLM scores one state significantly higher than the other, and the cumulative confidence score of all states is high, it is adopted as the pseudo-label. Otherwise, it is marked as ambiguous. See Fig. 2a (right) for an illustration. This process generates preliminary pseudo-labels which we refine further as described next.

3.3. OSC Dynamics Constraints

While CLIP-based similarity matching provides an initial set of pseudo-labels to bootstrap training, these labels contain considerable noise. For example, labels for a specific object may erroneously transition from transformed to actionable state, or regions initially marked as background may later be relabeled as the object of interest. To effectively train our model for spatially-progressing object state change segmentation, we first refine these pseudo-labels using a set of state-change dynamics constraints that capture the core aspects of state transitions.

We first define the variables in use. Let $c \in \{s_{\text{act}}, s_{\text{trf}}, s_{\text{amb}}\}$ denote the object-state class. Given a video with T frames, for each mask proposal k at time-step t, let l_t^k represent the pseudo-label. We define $\mathbb{S}_c^k = \{t \in [1, T] : l_t^k = c\}$ as the set of time indices where proposal k is labeled as c. With masklets corresponding to individual

[2] Our inference formulation allows predicting "background" labels.

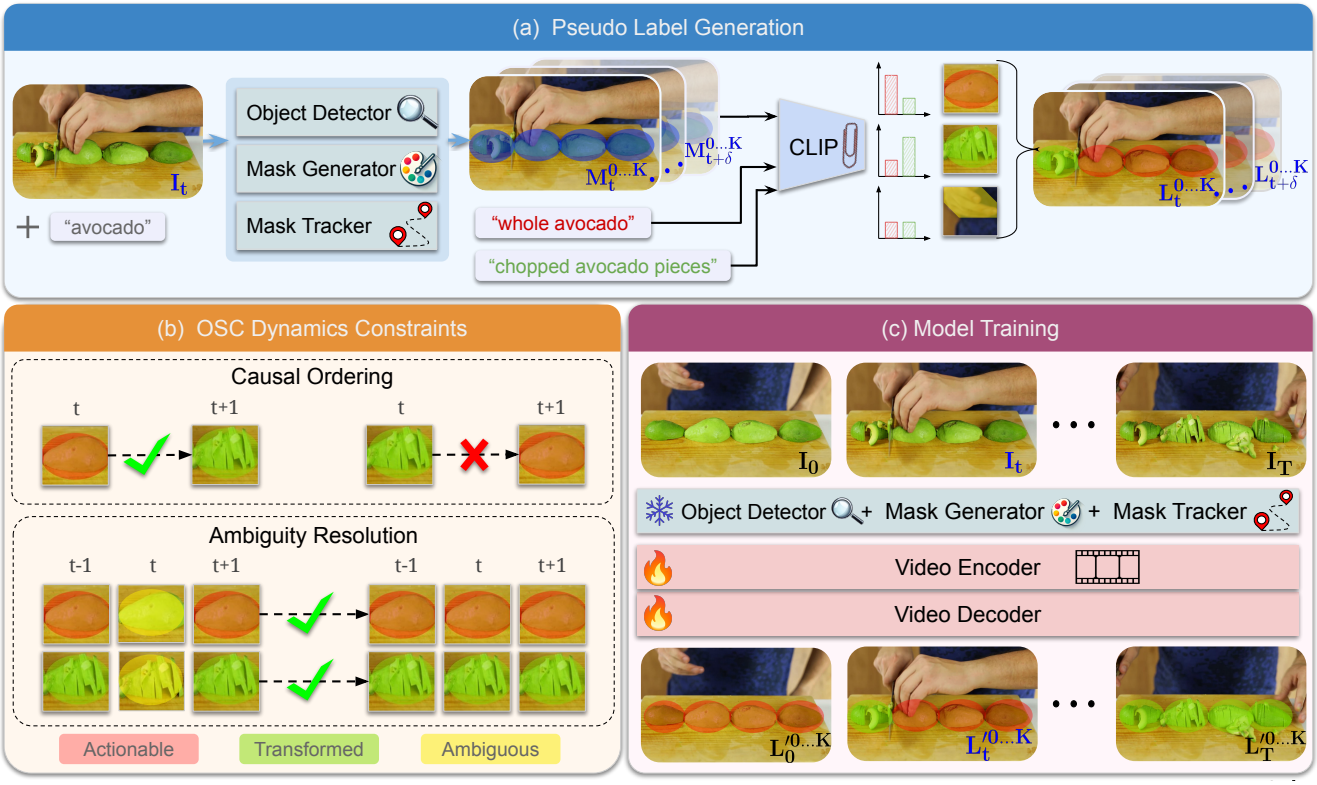

Figure 2. **Overview of SPOC. a)** Pseudo-label generation (Sec. 3.2): Given a video of a human performing a state-changing activity, we use off-the-shelf object detection [18], mask generation [14] and tracking models [43] to extract a set of region mask proposals $M_t^{0\ldots K}$ for each frame I_t. We then use CLIP [32] to apply similarity-score matching of visual region embeddings with textual state-description embeddings to obtain max-similarity pseudo-labels L_t^k for each region. **b)** OSC dynamics constraints (Sec. 3.3): We refine the pseudo-labels by incorporating several important dynamics constraints that emphasize the temporal progression of state-change transitions while respecting their causal dynamics. **c)** Model training (Sec. 3.4): Using our large-scale pseudo-labeled dataset, we train a video model to classify mask proposals into one of three classes: *actionable*, *transformed*, or *background*.

object instances, this formulation allows us to apply state-change dynamics constraints on pseudo-labels at the object-level, extending beyond frame-level application [36].

Causal ordering We define a causality constraint to enforce a coherent progressive order among the labels at the *object-level*. For a given tracked masklet k, the sequence of states should adhere to a progression where any instances of transformed states s_{trf} appear after actionable states s_{act}. To enforce this constraint, we first compute their time-series mid-points:

$$\text{mid}_{\text{act}} = \frac{\sum_{t \in \mathbb{S}^k_{\text{act}}} t}{|\mathbb{S}^k_{\text{act}}|}, \quad \text{mid}_{\text{trf}} = \frac{\sum_{t \in \mathbb{S}^k_{\text{trf}}} t}{|\mathbb{S}^k_{\text{trf}}|}$$

If $\mathbb{S}_{act}[-1] > \mathbb{S}_{trf}[0]$ (i.e., actionable indices follow transformed), we compute $dist(\mathbb{S}_{act}[-1], mid_{act})$ and $dist(\mathbb{S}_{trf}[0], mid_{trf})$. If the former is greater (indicating a potential mislabeled outlier in $\mathbb{S}_{act}$), we flip $\mathbb{S}_{act}[-1]$; else flip $\mathbb{S}_{trf}[0]$. We iteratively repeat until $\mathbb{S}_{act}[-1] < \mathbb{S}_{trf}[0]$. After this procedure, all indices in $\mathbb{S}_{act}$ precede $\mathbb{S}_{trf}$, enforcing a cohesive causal ordering in the pseudolabels.

Ambiguity Resolution After enforcing causal ordering, we are still left with ambiguous pseudo-labels. For any ambiguous state s_{amb} in the sequence, we assign it to either transformed or actionable class based on proximity:

$$l_t^k \in s_{\text{amb}} \rightarrow \begin{cases} s_{\text{act}}, & \text{if } |t - \max(S^k_{\text{act}})| < |t - \min(S^k_{\text{trf}})| \\ s_{\text{trf}}, & \text{otherwise} \end{cases}$$

where t is the frame index with an ambiguous label. This re-labeling maintains continuity in state transitions, even for labels beyond the actionable-transformed boundaries.

By refining the label sequences with these dynamics constraints, we significantly reduce label noise, creating a more stable and logically consistent training set. This label refinement process—which goes beyond traditional temporal smoothing—plays an important role in enhancing the reliability of pseudo-labels, enabling our model to learn from a structured representation of object state dynamics.

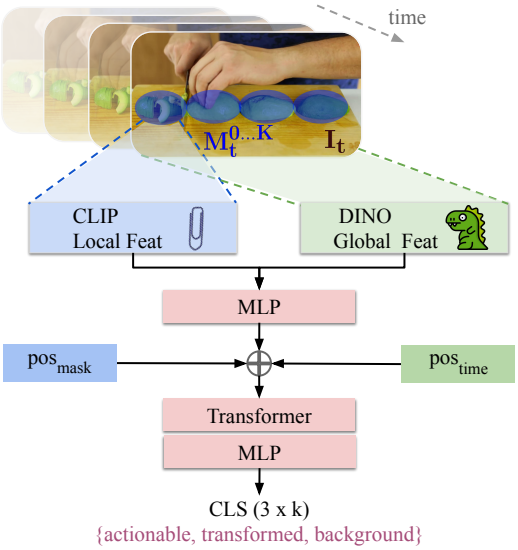

Figure 3. Model architecture (Sec. 3.4)

3.4. Model Training

We now describe the components involved in training our model on the generated pseudo-labels. The model architecture (Fig. 2c) consists of a video encoder to process input frames and a decoder to predict state labels per mask proposal. Each of the components are expanded in Fig. 3. The input to the model are video frames $\{I_0, I_t, \ldots, I_T\}$ and corresponding object masks $\{M_t^{0..K}\}$. We then extract local-global features for each mask and image frame using CLIP [32] and DINOv2 [27] respectively. While the former captures fine-grained mask-level state information, the latter provides frame-level global context. We aggregate the two features and feed them through a transformer, using a combination of mask-positional pos_{mask} and time-positional embeddings pos_{time}. Finally the transformer is trained with cross-entropy loss to classify each region into {actionable, transformed, or background} across the video frames. By leveraging both CLIP and DINO, the model captures detailed object-level and contextual frame-level information, while the transformer models temporal dynamics essential for understanding state transitions.

4. The WhereToChange Dataset

Existing OSC datasets do not capture the fine-grained information necessary to evaluate segmenting spatially-progressing OSCs. To address this gap, we introduce WhereToChange, a large-scale dataset featuring detailed intra-object state-change annotations across a wide variety of objects and actions. Below, we provide details on data collection and annotation with full details in Appendix B.

Data Collection We source our dataset from the recent HowToChange dataset [42]. Derived from HowTo100M [23], which comprises a vast collection of YouTube videos, HowToChange provides frame-level state labels for cooking activity videos. In building WhereToChange (WTC for short), we focus on 10 state transitions from HowToChange that exhibit spatial progression behavior (see Table 2). Using the pseudo-labeling process (Sec. 3.2), we construct a training set of ∼17k videos, encompassing ∼700k frames, with an average clip duration of 41 seconds. All evaluation sets are manually labeled (see Sec. 5.1). WhereToChange includes 153 OSCs spanning 73 objects labeled as seen across train and test, and 60 OSCs spanning 43 objects labeled as novel during testing.

Datasets	# Obj	# ST	# OSC	# Videos (train / eval)	Seg	OSC Label	Intra-object
ChangeIt [36]	42	27	44	34,428 / 667	✗	✓	✗
HowToChange [42]	134	20	409	34,550 / 5,424	✗	✓	✗
VSCOS [45]	124	30	271	1,905 / 98	✓	✗	✗
VOST [39]	57	25	51	713 / 141	✓	✗	✗
WhereToChange (ours)	116	10	213	16,999 / 1,162	✓	✓	✓

Table 1. **Comparison with existing video OSC datasets.** 'Obj' and 'ST' represent objects and state transitions, respectively. We present the first benchmark for spatially-progressing video OSC, with fine-grained intra-object state-change segmentations. Among segmentation datasets, we offer unparalleled scale, while significantly enhancing state-change granularity.

Table 1 compares our dataset with existing video datasets on OSC understanding. WhereToChange is the first benchmark for spatially-progressing video OSC, pioneering fine-grained intra-object state-change segmentations—hitherto unexplored in prior work. Among segmentation datasets, it offers unparalleled scale, featuring 9× more training data and 8× more evaluation data compared to existing datasets [39, 45]. This significant expansion enhances both the granularity and utility of OSC annotations within individual frames, setting a new standard for the field.

5. Experiments

We define datasets, metrics, baselines then present results.

5.1. Setup

Datasets For evaluation, we obtain ground truth human annotations for video clips from two datasets: WTC-HowTo and WTC-VOST. While the former is sourced from large-scale YouTube instructional videos [23] of free-form, mostly third-person videos with diverse objects, the latter [39] consists of continuously captured human activity from egocentric video datasets Ego4D [11] and EPIC-Kitchens [8]. Within VOST, we consider two spatially-progressing OSC categories: chop and peel. The combination of datasets provides a comprehensive testbed, and allows testing our HowTo-trained model on a distinct dataset.

Model	WTC-HowTo											WTC-VOST		
	Mean	Chop	Crush	Coat	Grate	Mash	Melt	Mince	Peel	Shred	Slice	Mean	Chop	Peel
w/o training														
MaskCLIP [49]	0.146	0.165	0.095	0.134	0.173	0.163	0.107	0.110	0.154	0.206	0.153	0.098	0.111	0.085
GroundedSAM [34]	0.273	0.286	0.235	0.275	0.317	0.272	0.173	0.318	0.256	0.310	0.284	0.221	0.271	0.172
SAM-Track [7]	0.275	0.289	0.216	0.323	0.276	0.311	0.198	0.286	0.283	0.294	0.272	0.194	0.227	0.161
Random Label	0.274	0.278	0.242	0.279	0.269	0.303	0.275	0.272	0.259	0.295	0.269	0.184	0.190	0.179
DEVA [5]	0.282	0.321	0.198	0.256	0.333	0.289	0.223	0.328	0.211	0.370	0.293	0.228	**0.298**	0.157
SPOC (PL)	0.411	0.438	0.351	0.237	0.429	0.527	0.416	0.493	0.373	0.418	0.428	0.204	0.229	0.179
+CO	0.438	0.456	0.371	0.378	0.441	0.553	0.415	0.497	0.380	0.442	0.445	0.216	0.241	0.191
+AR	**0.455**	**0.461**	**0.383**	**0.447**	**0.447**	**0.566**	**0.418**	**0.500**	**0.417**	**0.453**	**0.458**	**0.229**	0.257	**0.201**
w/ training														
MaskCLIP+ [49]	0.182	0.200	0.125	0.168	0.216	0.197	0.138	0.143	0.194	0.256	0.186	0.119	0.136	0.103
GroundedSAM [34]	0.275	0.291	0.225	0.282	0.323	0.274	0.171	0.325	0.259	0.313	0.285	0.223	0.275	0.171
SPOC	**0.502**	**0.523**	**0.422**	**0.526**	**0.528**	**0.610**	**0.425**	**0.541**	**0.449**	**0.503**	**0.494**	**0.267**	**0.307**	**0.227**

Table 2. **Results on WhereToChange-HowTo and WhereToChange-VOST.** Comparison of different models based on mIoU performance. Our pseudo-labeling strategy (Sec. 3.2: PL=pseudo-labels) substantially outperforms existing state-of-the-art object segmentation baselines, with dynamics constraints (Sec. 3.3: CO=causal-ordering, AR=ambiguity-resolution) further enhancing performance. The fully trained model (Sec. 3.4) achieves the highest accuracy, benefiting from large-scale training. Notably, our model trained on HowTo also outperforms the baselines on VOST, a challenging out-of-distribution dataset comprising continuously captured egocentric videos.

Professional annotators manually annotate 1001 and 155 videos from HowToChange and VOST to form our evaluation sets WTC-HowTo and WTC-VOST, respectively. Each annotator is presented with frames from a video clip and the corresponding OSC category. Their task is to label pixels on the object of interest in each frame as either actionable or transformed. Clips with ambiguity in distinguishing between states are identified and excluded from the evaluation set, ensuring high quality. The annotation process was conducted on TORAS [13] with the assistance of 7 human annotators, totaling 350 hours.[3]

Metrics and Data Splits We report Jaccard Index ($\mathcal{J}$), also called Intersection over Union (IoU) between predicted and ground truth masks, computed as the mean IoU over actionable and transformed masks. Following [35, 39], we do not report a boundary $\mathcal{F}$-measure since object contours in OSCs are ill-defined due to drastic transformations over time.

In addition to the full test sets, we provide results on three distinct subsets of WhereToChange: WTC-Transition, focusing exclusively on transition frames from HowToChange (8405 frames, or 23%), and WTC-Seen/Novel, consisting of seen/novel object categories (novel = unseen during training). WTC-Transition isolates moments where the object undergoes significant state changes (excluding frames in the initial or end state), emphasizing the progressive spatial evolution of the object. WTC-Novel isolates the generalization ability to novel objects (e.g., seeing an apple being chopped at test time after only seeing other objects be chopped during training).

Implementation We sample videos at 1 fps for both training and annotation collection. We apply GroundingDino [18] (for object detection) and SAM [14] (for mask proposal generation) every 10 frames and track masks using DeAOT [43] at intermediate frames. Using ViT-B/32 as our CLIP [32] vision encoder for pseudo-labels, we set similarity thresholds in Eq. 1 using grid search. We train a separate model for each verb following prior work on OSC classification [36, 42]. Full details in Appendix C.1.

[3] Our training uses only pseudo-labels and does not require ground truth labels. The annotations are reserved exclusively for evaluation.

5.2. Baselines

Since no prior methods tackle the proposed task, we adapt existing high-quality open-vocabulary segmentation works for our task. They broadly fall under two categories:

(1) Pixel-based : Classify each pixel in the image into one of 3 categories—actionable, transformed, or background.

– **MaskCLIP** [49]: predicts dense segmentation by minimally adapting the final layers of CLIP.

– **MaskCLIP+** [49]: finetunes DeepLab-v2 [3] on the pseudo-labels generated by MaskCLIP on WTC-HowTo.

(2) Object-centric : Use open-vocabulary object detectors and segmentors to obtain relevant object regions and then classify them into the above 3 categories.

– **GroundedSAM** [34]: uses GroundingDino [18] to obtain text-prompted bounding boxes, which are then used to prompt SAM [14] for masks.

– **SAM-Track** [7]: incorporates DeAOT [43] into GroundedSAM [34] to propagate masks through time.

– **Random Label**: a naive baseline that randomly assigns actionable or transformed labels to SAM-Track masks.

– **DEVA** [5]: a decoupled video segmentation approach that uses XMem [4] to store and propagate masks.

Through the above baselines, we aim to comprehensively benchmark prior state-of-the-art open-vocabulary video object segmentation methods on our novel task. We finetune

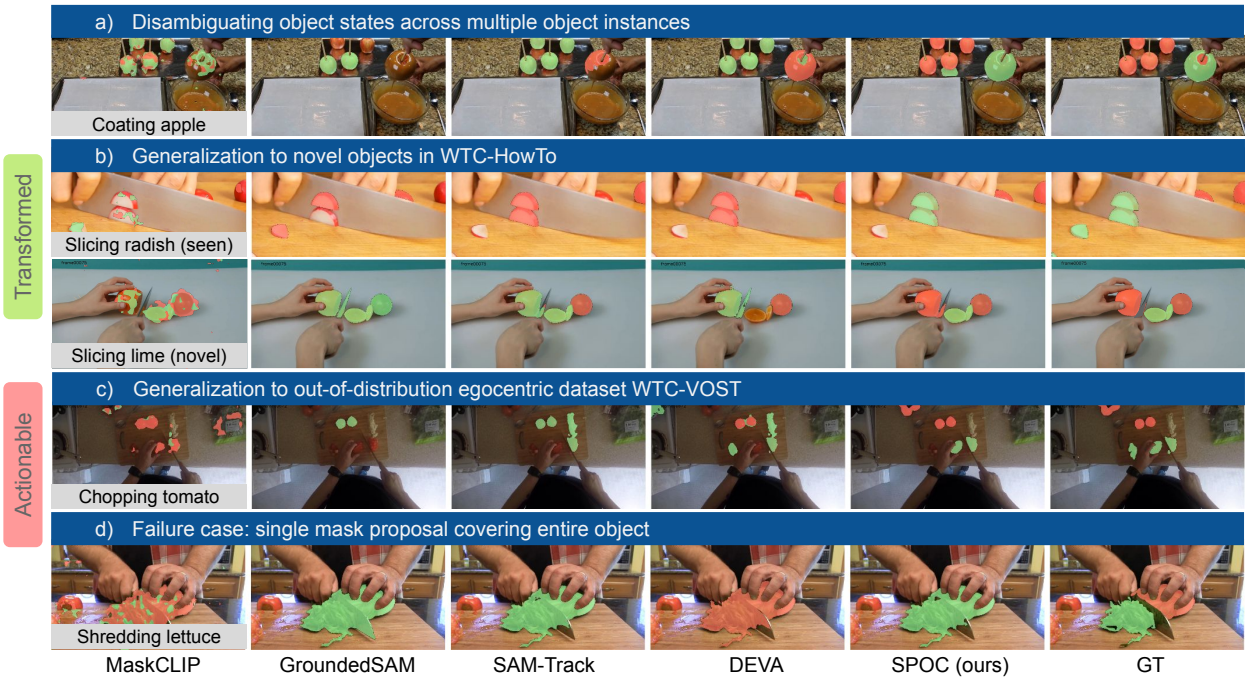

Figure 4. **Qualitative results.** a) SPOC clearly distinguishes between actionable and transformed instances of the state-changing object (coated vs uncoated apple), b) with the ability to generalize to novel unseen objects (slicing lime). In contrast, baseline methods tend to be state-change agnostic with decreased ability to disambiguate object states. c) SPOC also shows good generalization to the challenging out-of-distribution VOST dataset. d) Failure cases arise from singular mask proposals spanning the entire object during transitions (single mask for the full lettuce), affecting the model's intra-object segmentation capability.

Model	w/o OSC	Full	Transition	Seen	Novel
MaskCLIP [49]	0.346	0.146	0.152	0.148	0.144
MaskCLIP+ [49]	0.393	0.182	0.181	0.188	0.184
GroundedSAM [34]	0.611	0.273	0.262	0.276	0.270
SAM-Track [7]	0.534	0.275	0.240	0.279	0.270
DEVA [5]	0.519	0.282	0.269	0.283	0.281
SPOC (ours)	0.555	**0.502**	**0.332**	**0.523**	**0.492**

Table 3. **Performance across different data splits of WTC-HowTo.** Owing to large-scale training, SPOC shows strong generalization to novel objects.

baselines on WTC-HowTo where feasible (MaskCLIP+, GroundedSAM). Details in Appendix C.2.

5.3. Results

In our experiments, we aim to answer six key questions:

Can SPOC segment spatially-progressing OSCs? Table 2 shows SPOC outperforms all baselines. The object-centric methods outperform the pixel-based approaches, which tend to introduce higher noise in dense predictions. However, their performance remains close to the naive Random Label baseline, highlighting the difficulty of fine-grained state distinction. Our pseudo-labeling strategy substantially outperforms existing segmentation baselines, with the trained model further improving performance. From qualitative results shown in Fig. 4, our method clearly distinguishes between actionable and transformed instances of the same object. E.g., for coating apple, where multiple apples transition between uncoated and coated states, SPOC accurately classifies each apple by its state. In contrast, baseline methods—even when fine-tuned with our same training videos—are generally state-agnostic, with decreased ability to disambiguate between the two states. Failure cases (Fig. 4d) typically arise when single mask proposals cover the entire object during transitions, limiting the model's effectiveness in achieving fine-grained intra-object segmentation (Table 3-Transition).

Is spatially segmenting OSCs more challenging than segmenting the object itself? To further evaluate the complexity of this task, we assess performance in predicting the full object segmentations on the same test videos, i.e., by fusing the actionable/transformed masks into a single, state-agnostic object mask (Table 3-w/o OSC). Notably, these results are considerably better than those for fine-grained classification into 'actionable' and 'transformed' states. This shows that existing models struggle with this fine-grained state distinction, *even when they can capture the object relatively well.*

Does SPOC generalize to novel OSCs? We retain the seen/novel object splits from HowToChange and evaluate generalization to novel objects on WTC-HowTo (Table 3). Our large-scale training enables strong generalization, high-

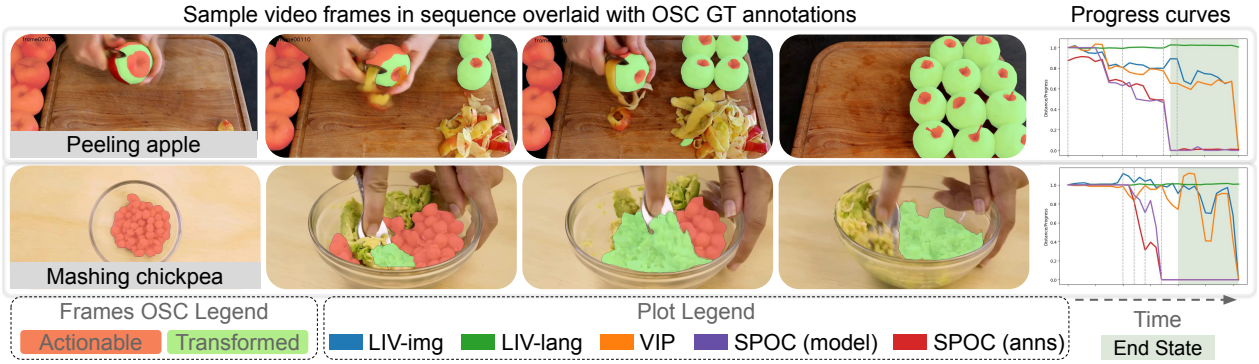

Figure 5. **Activity progress curves.** We show sample frames from a video sequence with progress curves generated by different methods, where vertical lines indicate the time-steps of sampled frames. Ideal curves should decrease monotonically, and saturate upon reaching the end state. In contrast to 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.

Model	**WTC-HowTo**			**WTC-VOST**
	$\tau \uparrow$	$end_\sigma \downarrow$	$end_{l2} \downarrow$	$\tau \uparrow$
VIP [20]	-0.437	0.167	0.639	-0.405
LIV-lang [21]	-0.031	-	0.999	-0.062
LIV-lang (finetune) [21]	0.054	-	1.005	0.048
LIV-img [21]	-0.468	0.114	0.644	-0.401
LIV-img (finetune) [21]	-0.418	0.113	0.673	-0.374
SPOC (model)	**-0.611**	**0.057**	**0.212**	**-0.581**
SPOC (annotations)	-0.670	0.031	0.206	-0.651

Table 4. **Activity progress metrics.** OSC-based curves outperform state-of-the-art representation learning methods for robot-learning. τ represents Kendall's Tau as the monotonicity index; end_σ and end_{l2} are task completion metrics.

lighting the role of high-quality pseudo-labels in enhancing performance. E.g., the model learns concepts like "sliced pieces" during training, allowing it to segment novel objects into whole and sliced pieces during testing (Fig. 4b).

Does SPOC generalize to new datasets? From our results on VOST in Table 2, the SPOC model trained on HowTo achieves the best overall performance on this challenging dataset of out-of-distribution, continuously captured egocentric videos (Fig. 4c). However, a primary limitation in this dataset was unreliable mask tracking of SAM-Track [7], the off-the-shelf tracker we employed, which struggled with the sudden, jerky head movements common in egocentric video. Future advancements in object detection and mask tracking are likely to further enhance our model's performance on these complex video sequences.

How important are the state-change dynamics constraints? To test the efficacy of our state-change dynamics constraints (Sec. 3.3), we progressively incorporate each component and evaluate performance in Table 2. Starting with the CLIP-similarity-based pseudo-labels, each additional dynamics constraint markedly improves segmentation, culminating in an overall improvement of 22%.

How could intra-object spatially-progressing OSCs be used for robotics? A primary motivation for this novel task is to advance activity progress monitoring, vital for AR/MR and robotics. To test this, we conduct initial experiments applying SPOC estimates as a continuous task-progress reward for robot learning, formulated in terms of area of object state change: $R_t = \frac{A_t^{act}}{A_t^{act} \cup A_t^{trf}}$, where A_t^{act} and A_t^{trf} represent the areas of actionable and transformed regions at time step t. We compare against LIV [21] and VIP [20], state-of-the-art methods that learn goal-based representations for robots from human-activity videos [8, 11].

Fig. 5 shows reward curves for WhereToChange clips. Following the paradigm of [20, 21], ideal curves should decrease monotonically, and saturate at the end state. Our OSC curves accurately reflect task progress and provide a continuous reward formulation, crucial for robot learning. Quantitative metrics (described in Appendix E) on monotonicity and task completion in Table 4 show that SPOC significantly outperforms baselines. These results underscore the downstream utility of our proposed task: fine-grained intra-object state-change segmentation enables real-time feedback for robots.

6. Conclusion

We introduce a novel task for video OSC understanding through spatially-progressing OSC segmentation. Our large-scale WhereToChange dataset spans diverse activities, offering unmatched granularity. Leveraging vision-language models and our causality and temporal progression constraints, we demonstrate our method's efficacy in segmenting and tracking object state changes. Our model outperforms baselines, generalizes to novel objects, and captures dynamic state changes, setting a new benchmark for spatio-temporal OSC segmentation with implications for activity tracking in AR/MR and robot learning.

References

[1] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2127–2136, 2017. 2

[2] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 2

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 6

[4] Ho Kei Cheng and Alexander G. Schwing. XMem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 2, 6, 20

[5] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 6, 7, 19, 20

[6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[7] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2, 6, 7, 8, 19

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018. 3, 5, 8

[9] Gerard Donahue and Ehsan Elhamifar. Learning to predict activity progress by self-supervised video alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 22

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 22

[11] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 3, 5, 8, 21

[12] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2

[13] Amlan Kar, Seung Wook Kim, Marko Boben, Jun Gao, Tianxing Li, Huan Ling, Zian Wang, and Sanja Fidler. Toronto annotation suite. https://aidemos.cs.toronto.edu/toras, 2021. 6, 11, 12

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 4, 6, 19, 20

[15] Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9326–9335, 2022. 2

[16] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[17] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[18] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *European Conference on Computer Vision (ECCV)*, 2024. 3, 4, 6, 19, 20

[19] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly recognizing object fluents and tasks in egocentric videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2924–2932, 2017. 2

[20] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022. 8, 21, 22, 24

[21] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023. 8, 21, 22, 24

[22] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5222–5230, 2021. 2

[23] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 3, 5

[24] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1792–1801, 2017. 2

[25] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 953–962, 2021.

[26] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018. 2

[27] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 5, 20

[28] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[29] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 2, 20

[30] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13018–13028, 2021. 2

[31] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602, 2019. 2

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4, 5, 6, 19, 20

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[34] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024. 2, 3, 6, 7, 19, 20

[35] Yuhan Shen, Huiyu Wang, Xitong Yang, Matt Feiszli, Ehsan Elhamifar, Lorenzo Torresani, and Effrosyni Mavroudi. Learning to segment referred objects from narrated egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6, 20

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 4, 5, 6, 14

[37] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Multi-task learning of object state changes from uncurated videos. *TPAMI*, 2024. 1, 2

[38] Tomáš Souček, Dima Damen, Michael Wray, Ivan Laptev, and Josef Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[39] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the "object" in video object segmentation. In *CVPR*, 2023. 1, 2, 5, 6, 11, 14, 18, 20

[40] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models. *arXiv preprint arXiv: 2303.04803*, 2023. 2

[41] 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 *Proceedings of the European conference on computer vision (ECCV)*, 2018. 2

[42] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 5, 6, 11, 14, 16, 17

[43] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 4, 6, 18, 19

[44] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[45] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 5, 14, 20

[46] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20439–20448, 2023. 2

[47] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *NeurIPS*, 2023. 2

[48] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[49] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 6, 7, 19

Appendix

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 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 [42], 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 ($\sim$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 [42] (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 [39], 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 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.

ii) Annotation User Interface

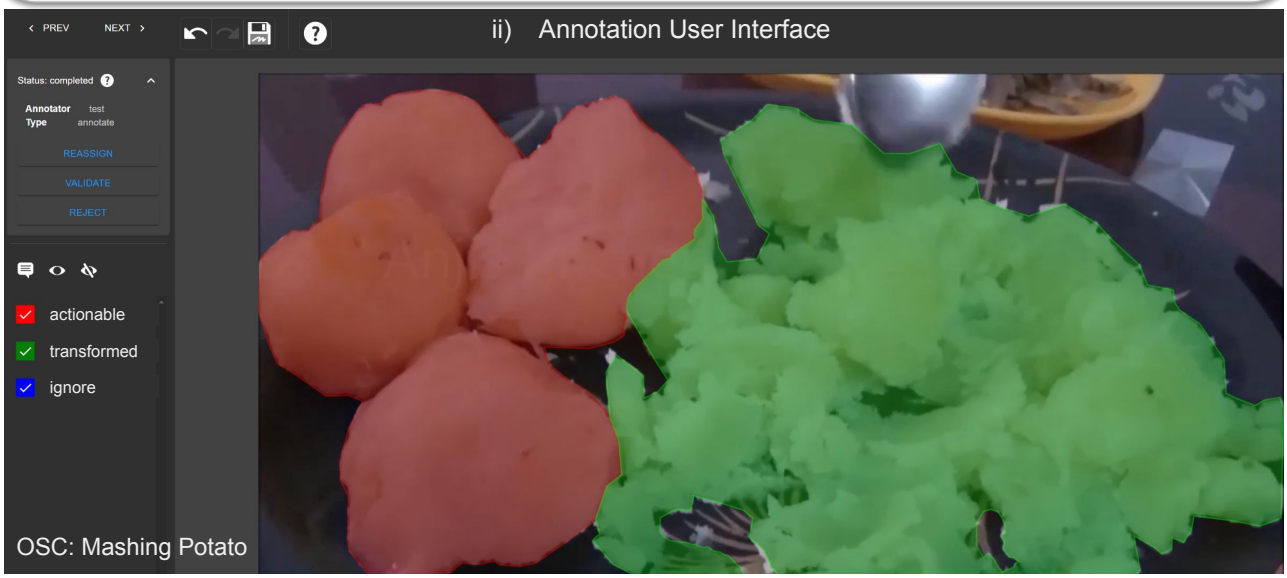

Figure A. **Annotation User Interface and Guidelines.** **(i)** Overview of general annotation guidelines provided to the annotators. Verb-specific guidelines are in Table B. **(ii)** Annotation user interface on TORAS [13]. Details in Sec. B.2.

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.

vided 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 proceed 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 WhereToChangedataset 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.

Verb	Guidelines
Chopping	*Actionable*: Whole fruits and large pieces *Transformed*: 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	*Actionable*: Whole or chunked piece *Transformed*: Grated pieces Note: Grater and other tools are ignored
Shredding	Same as above
Mashing	*Actionable*: Whole fruit or large chunks *Transformed*: 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	*Actionable*: Unpeeled region *Transformed*: 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	*Actionable*: Unmelted solid region *Transformed*: Melted liquid or paste-like region
Coating	*Actionable*: Object region plain and uncoated by external substance *Transformed*: 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).

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 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 a into 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 [36, 42] or state-agnostic segmentations in [39, 45], 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

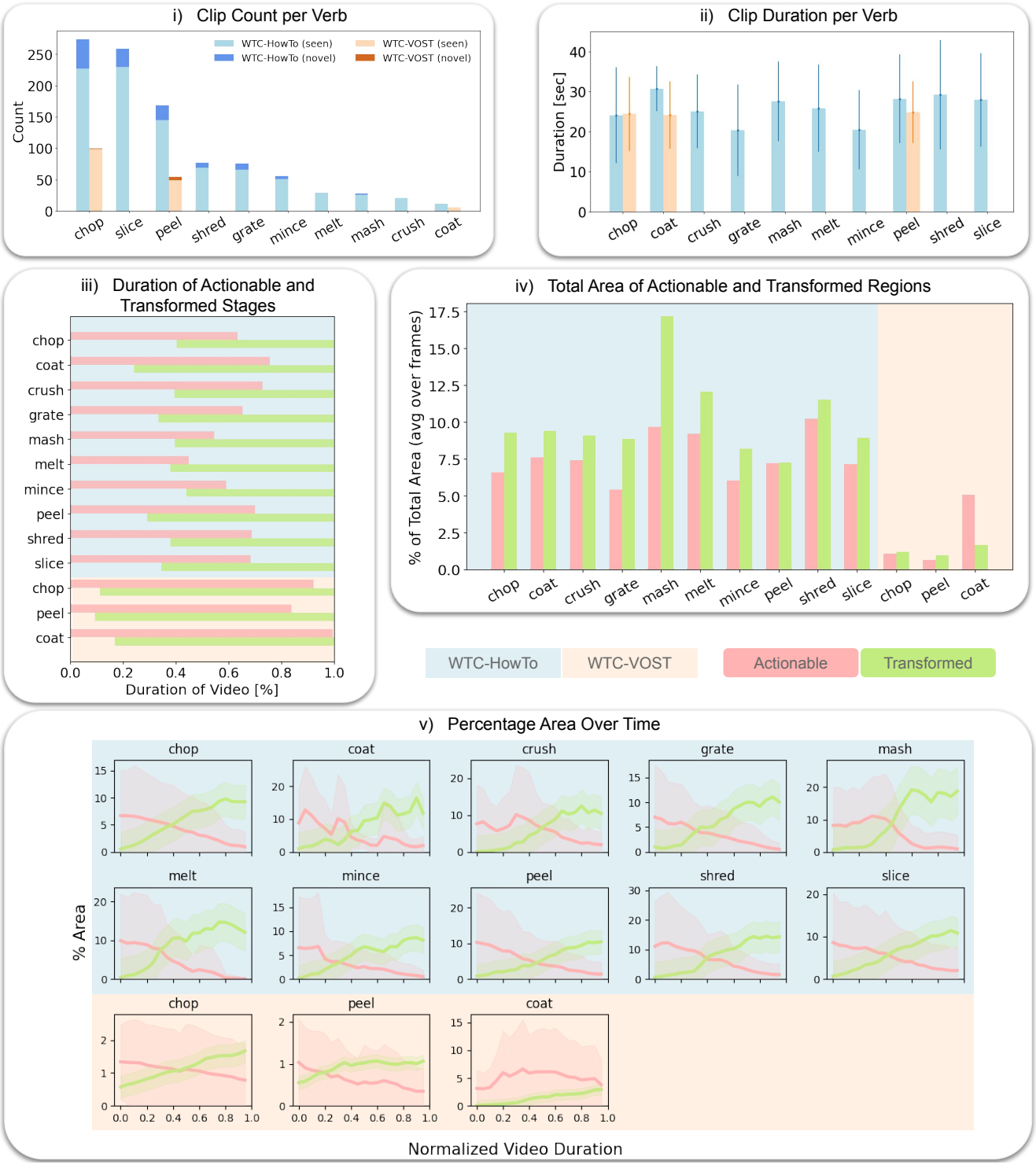

Figure B. **Analysis of WhereToChange dataset along different axes. i) Clip counts per verb:** distribution of seen-novel object counts in WTC-HowTo and WTC-Vost. Object taxonomoy in Table A. **ii) Clip duration per verb:** clips are 20 seconds long on average across all verbs. **iii) Duration of actionable 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.

Figure C. **Annotation samples from WhereToChange-HowTo.** Diverse frame samples from the HowToChange [42] 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 [42] 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.

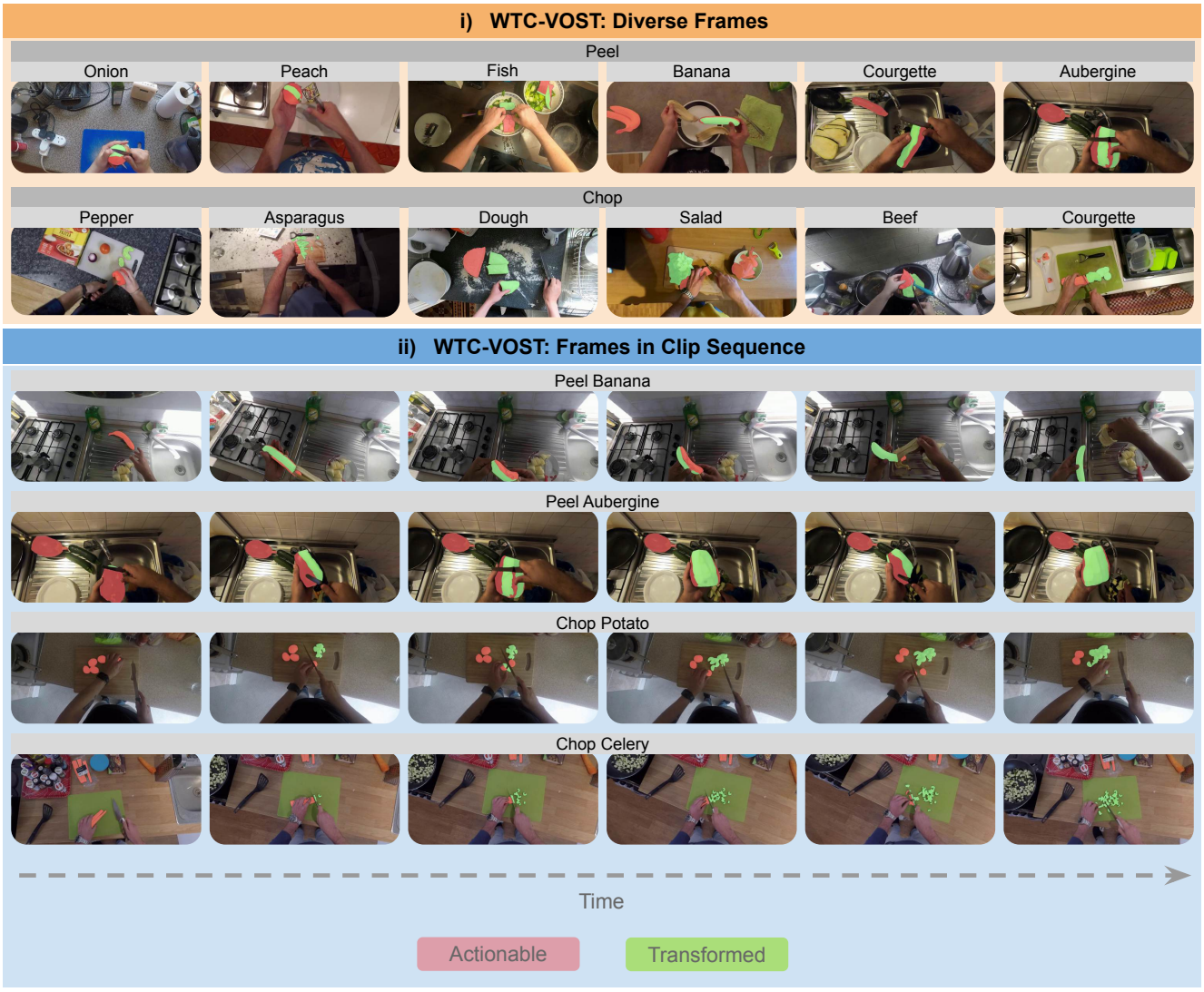

Figure E. **Annotation samples and clip sequences from WhereToChange-VOST.** We show sample frames **(i)** and clip sequences **(ii)** from the VOST [39] subset of our proposed WhereToChange dataset. This is a more challenging out-of-distribution dataset comprising continuously captured egocentric videos of human activities.

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.

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 [43], 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.

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 manual examples (e.g. prompt: "OSC: chopping avocado, ac-

tionable: 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.

Hyperparam	Value
Grounding-Dino [18]	
box threshold	0.35
text threshold	0.5
box size threshold	0.7
SAM [34]	
model type	vit_h
prompt type	bounding box
Clip [32]	
model	ViT-B/32
DeAOT [43]	
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[49], MaskCLIP+[49]) and object-centric (GroundedSAM[34], 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 [49] 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+ [49] 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 [34] 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 relevant objects, this method also uses DeAOT [43], 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 aggragated 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.

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 [29, 34, 39, 45], we report mean IoU scores as a measure of segmentation performance. The reported mIoU is the average over actionable and transformed regions:

$$mIoU = \frac{mIoU_{act} + mIoU_{trf}}{2} \tag{2}$$

Following prior segmentation works [35, 39] 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 [32] 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 $\Sigma s_{act}, s_{trf}$ $\Delta s_{act}, s_{trf}$ 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. F shows a heatmap of $\sum(S_{act}, S_{trf})$ 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.

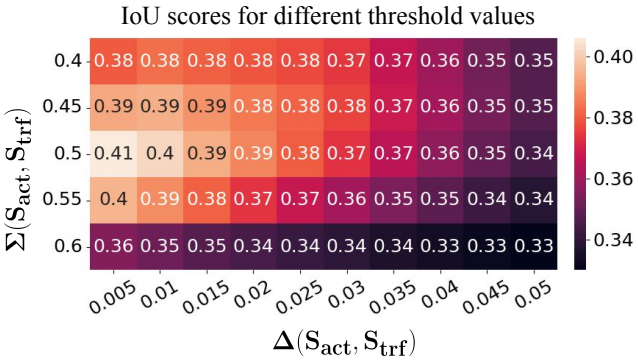

Figure F. **IoU scores for different CLIP similarity threshold values.** We plot a heatmap of $\sum(S_{act}, S_{trf})$ and $\Delta(S_{act}, S_{trf})$, CLIP similarity threshold values in Eq. 1. This heatmap shows aggregate values across all verbs. For our pseudo-labels, we choose best threshold values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

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. We mitigate this issue 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 [27] features and local mask-level CLIP [32] 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

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.

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.

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.

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. G. 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.

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. They 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:

$$R(t) = \Phi(s_t) - \Phi(s_g) \tag{3}$$

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 in our case.

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.

Model	WTC-HowTo										
	Mean	**Chop**	**Crush**	**Coat**	**Grate**	**Mash**	**Melt**	**Mince**	**Peel**	**Shred**	**Slice**
SPOC (pesudo-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.

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:

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

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 indicates 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. H. 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_{l2}) 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_{l2} values compared to the baselines. This is corroborated by the qualitative figures in Figs. 5 and H, 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. H: 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. I. 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

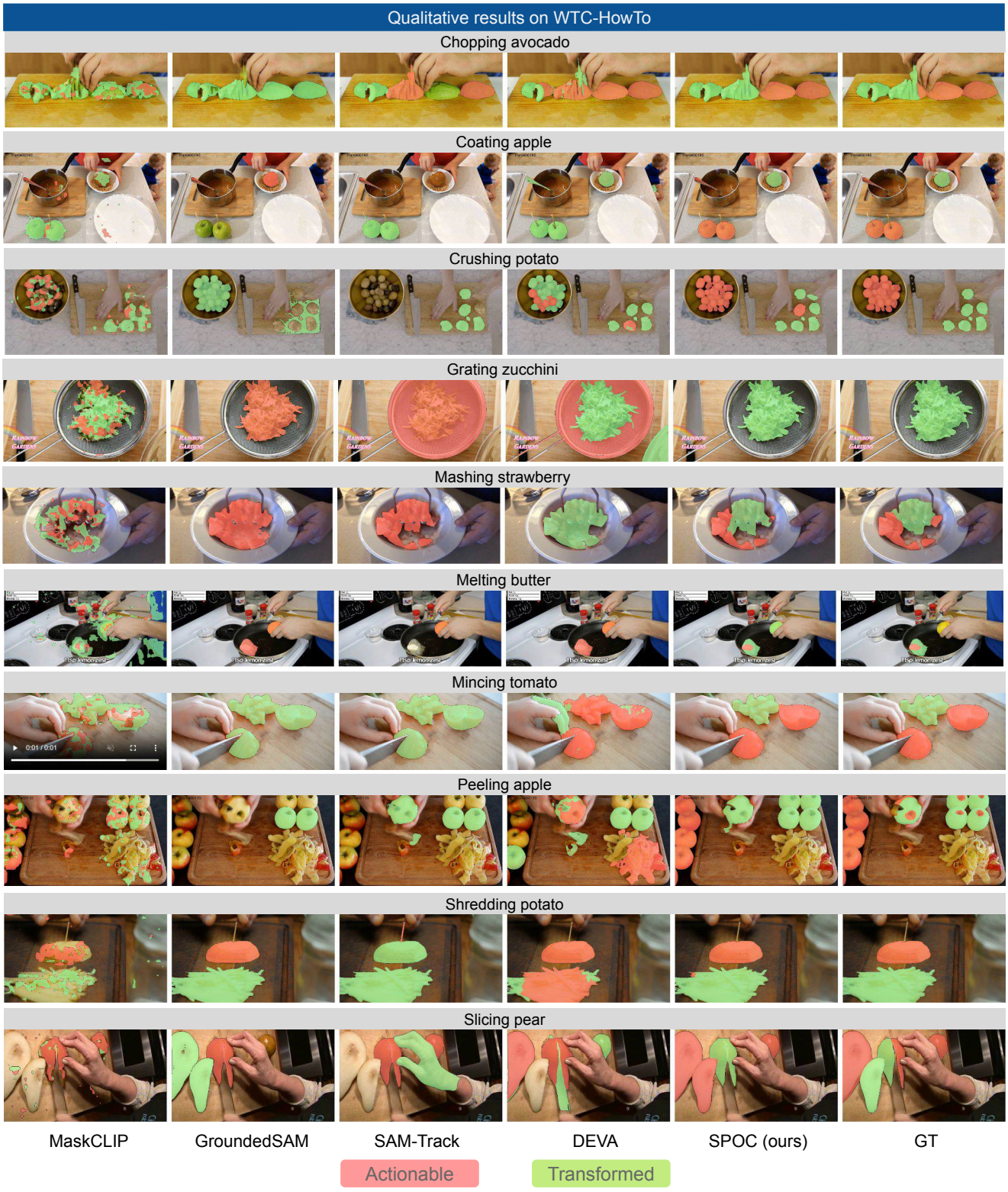

Figure G. **Qualitative results on WTC-HowTo subset.** In addition to the results in Fig. 4, we provide additional qualitative results comparing SPOC with all baselines across all verbs in WTC-HowTo. We observe that SPOC predictions are more sensitive to object states, while the baselines are more prone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

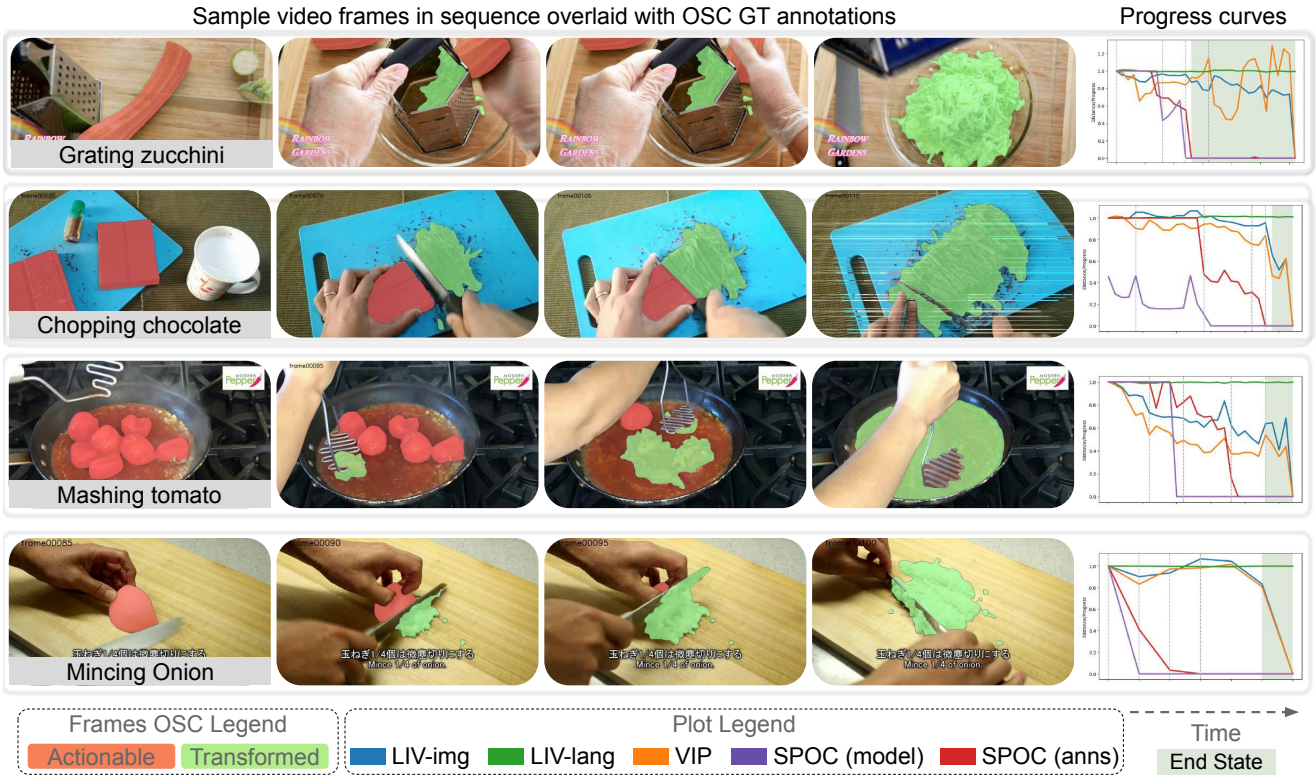

Figure H. **Activity progress curves.** Additional activity progress curves, like those in Fig. 5 in main. We show sample frames from a video sequence with progress curves generated by different methods, where vertical lines indicate the time-steps of sampled frames. Ideal curves should decrease monotonically, 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.

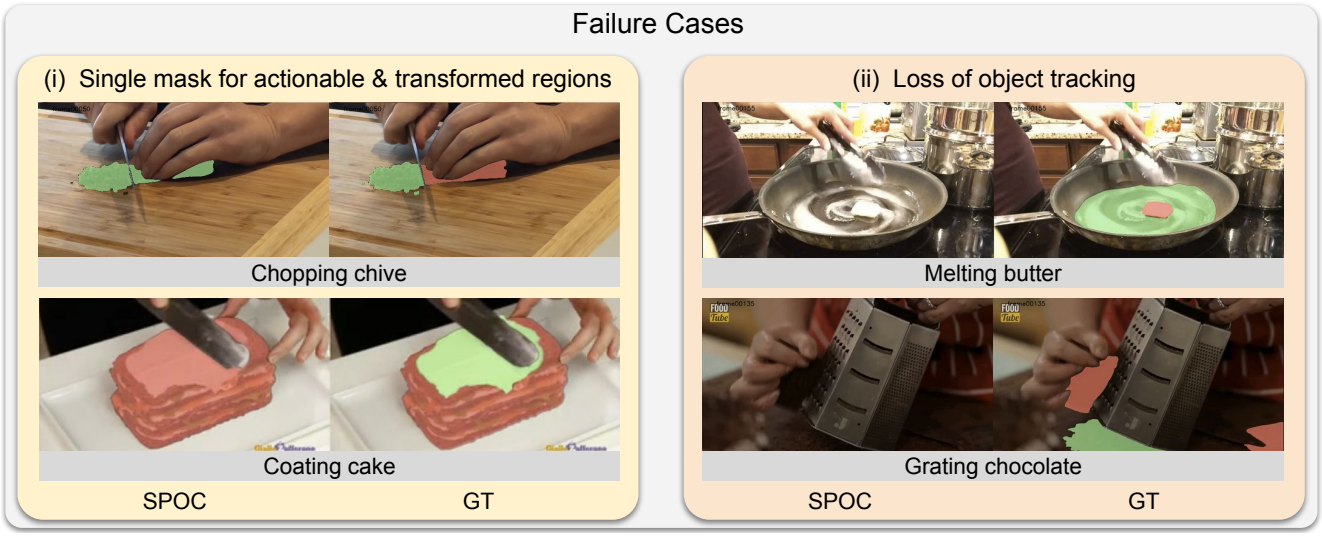

Figure I. **Failure modes.** We observe two main modes of failure in our model: **(i)** single mask proposal for both actionable and transformed regions (e.g. chopped and unchopped chive regions are within a single mask), and **(ii)** loss of object tracking (e.g. butter ceases to be tracked due to rapid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

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. I (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 timesteps. 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. I (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.

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.