1. Video Containing Qualitative Results

We invite the reader to view the video available at https://vision.cs.utexas.edu/projects/VidOSC/, where we provide: (1) a comprehensive overview of VidOSC, (2) video examples from HowToChange, and (3) qualitative examples of VidOSC's predictions. These examples highlight VidOSC's ability in identifying non-OSC moments as background and effectively distinguishing among the three OSC states. It delivers temporally smooth and coherent predictions that follow the natural OSC progression (from initial to transitioning, and then to the end state), and shows strong performance even with novel OSCs not seen in training. All these underscore the efficacy of VidOSC.

2. Video OSC

Expanding on Sec. 3.1, we clarify three aspects of our definition of video OSC. First, we focus on OSCs that lead to a visible change in an object's appearance. Processes that are non-visual or involve mere spatial movements (such as moving an apple from the sink to the cutting board) do not qualify as OSCs. Second, in line with previous works [2, 31, 50, 51], we operate under the assumption that each input video predominantly features a single OSC. The challenge of handling videos with multiple concurrent OSCs remains an intriguing avenue for future research. Lastly, during training, the input video and its OSC category name (e.g., shredding chicken) are available (as provided in both ChangeIt and our HowToChange, although not always accurate due to data collection noise). For evaluation, every test video (in both ChangeIt and HowToChange) is accompanied by a manually verified OSC category.

2.1. Data Collection

We streamline our dataset collection via an automated process. First we apply LLAMA2 [54] to ASR transcriptions in HowTo100M [36] with the following text prompt, one sentence at a time:

> *[Text Prompt to LLAMA2]* You will receive descriptions corresponding to a how-to instructional video. Your task is to identify any instances of Object State Change (OSC) based on the provided text. An OSC is a visually detectable transformation where an object undergoes a change that is difficult to reverse. Examples include apple peeling/cutting, bacon frying, milk boiling, butter melting, cake frosting, eggs whisking, cream whipping, etc.
>
> - Note 1: OSCs must be visually detectable. General actions like food preparing, or non-visual processes like onions sweetening, are not included.
> - Note 2: Simple spatial transitions, resulting from actions like add, mix, put, or place, are not considered OSCs.
> - Note 3: To qualify as an OSC, there must be a transition from one state to another, which should be indicated by an active action in the text description. The mere presence of a state (e.g., sliced pineapples, peeled apples) does not count unless there is explicit text describing the change (e.g., we slice the pineapples).
> - Note 4: Generally, sentences without OSCs are far more common than those with OSCs.
>
> To report identified OSCs, please use the following format: [object] + [state transition of the OSC]. Ensure the first word in each identified OSC is the object, and the subsequent words describe a state transition. If multiple OSCs are identified, separate them with semicolons (;). If no OSCs are detected, simply reply with None.

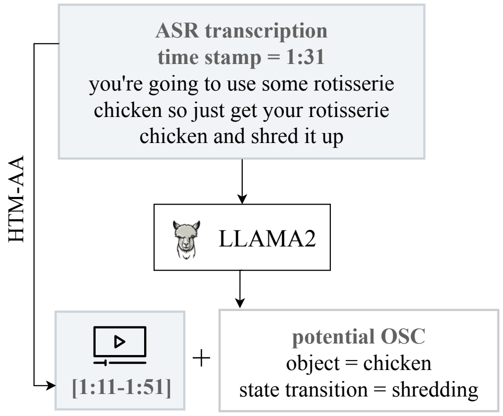

ASR transcription
time stamp = 1:31
you're going to use some rotisserie chicken so just get your rotisserie chicken and shred it up
HTM-AA
LLAMA2
[1:11-1:51]
+
potential OSC
object = chicken
state transition = shredding

Figure 7. Our proposed data collection process for HowToChange. HTM-AA [19] denotes the auto-aligned version of HowTo100M.

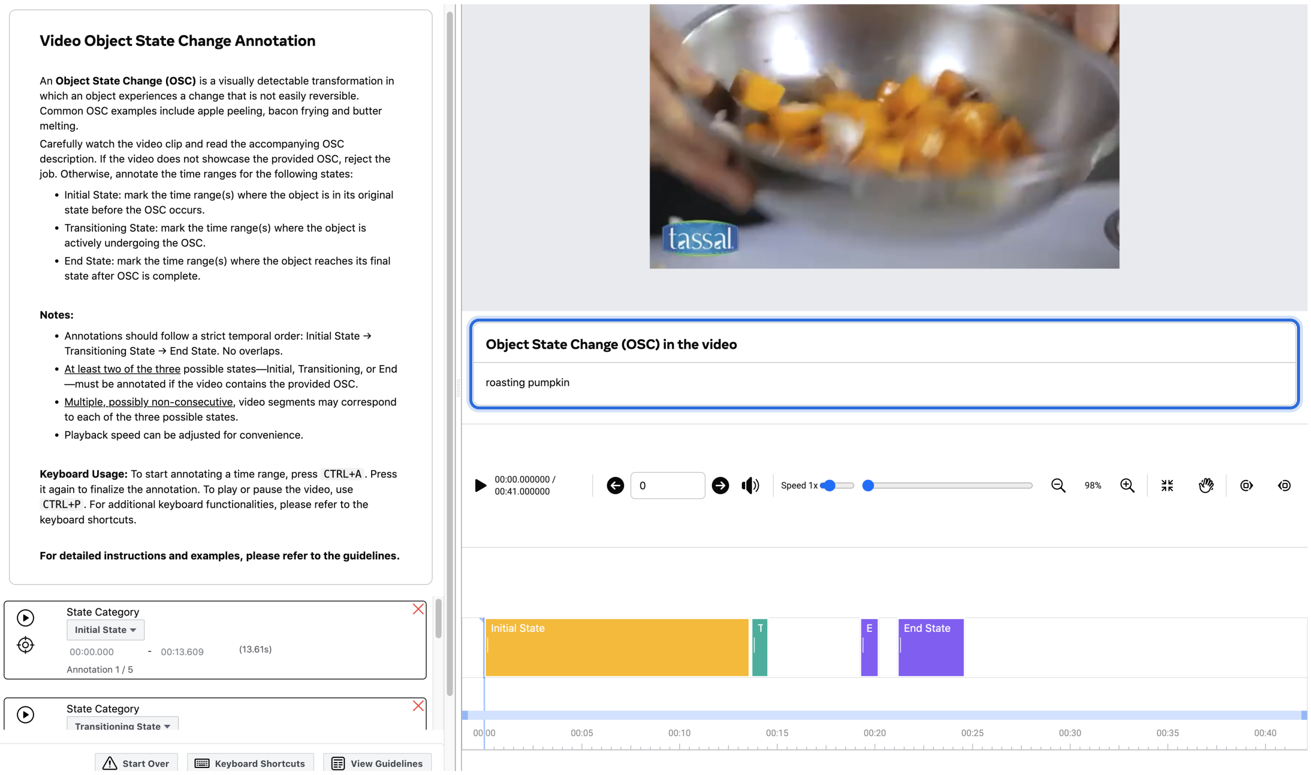
From the responses given by LLAMA2, we identify object and state transitions corresponding to the ASR transcription. Utilizing the HTM-AA dataset [19], where each ASR sentence corresponds to a time stamp in the video, we extract a clip centered around the identified time stamp with a ± 20 second window. The result is a cropped video segment of 40 seconds, paired with an OSC text (in the form of object + state transition). Finally, when multiple clips from the same video illustrate the same OSC with overlapping start and end times, we combine them into a single, extended clip. The whole process is illustrated in Fig. 7.

3. The HowToChangeDataset

To establish the OSC taxonomy, we identify 20 most frequent state transitions and the objects associated with these state transitions that appear more than 200 times. Utilizing a 0.25 quantile threshold for each state transition, we categorize the top 75% frequent OSCs as known and the bottom 25% as novel, resulting in 318 known and 91 novel OSC categories in total. See Table 4 for the complete OSC taxonomy. With these 318 known OSCs, we compose the training set of HowToChange, encompassing 36,076 video segments from HowTo100M. Fig. 9 provides the detailed distribution of HowToChange (Training).

3.1. Ground Truth Label Collection

For evaluation, we collect annotations from 30 trained professional human annotators for a subset of 5,423 video clips from HowToChange, amounting to 62.5 hours of video. See Fig. 8 for the annotation user interface. We collect an average of 13.3 annotated videos per OSC category. The annotations for known and novel



Video Object State Change Annotation

An **Object State Change (OSC)** is a visually detectable transformation in which an object experiences a change that is not easily reversible. Common OSC examples include apple peeling, bacon frying and butter melting.

Carefully watch the video clip and read the accompanying OSC description. If the video does not showcase the provided OSC, reject the job. Otherwise, annotate the time ranges for the following states:

- Initial State: mark the time range(s) where the object is in its original state before the OSC occurs.
- Transitioning State: mark the time range(s) where the object is actively undergoing the OSC.
- End State: mark the time range(s) where the object reaches its final state after OSC is complete.

Notes:

- Annotations should follow a strict temporal order: Initial State → Transitioning State → End State. No overlaps.
- At least two of the three possible states—Initial, Transitioning, or End—must be annotated if the video contains the provided OSC.
- Multiple, possibly non-consecutive, video segments may correspond to each of the three possible states.
- Playback speed can be adjusted for convenience.

Keyboard Usage: To start annotating a time range, press `CTRL+A`. Press it again to finalize the annotation. To play or pause the video, use `CTRL+P`. For additional keyboard functionalities, please refer to the keyboard shortcuts.

For detailed instructions and examples, please refer to the guidelines.

State Category
Initial State
00:00.000 - 00:13.609 (13.61s)
Annotation 1 / 5

State Category
Transitioning State

Object State Change (OSC) in the video

roasting pumpkin

00:00.000000 / 00:41.000000

Speed 1x

Figure 8. The annotation user interface. Annotators view a video paired with an OSC category, identified from the OSC mining process (outlined in Sec. 3.3). They are instructed to either reject the video if it does not demonstrate the specified OSC, or to annotate the time ranges corresponding to the initial, transitioning, and end states of the OSC shown in the video.

OSCs cap at 15 and 10, respectively. Fig. 10 provides the detailed distribution of HowToChange (Evaluation).

The breakdown of annotated time ranges within these videos is as follows: 19.8% for the initial state, 25.9% for the transitioning state, and 16.9% for the end state, with the remaining categorized as non-OSC-related background. Fig. 11 shows the distribution of video duration (seconds) and the number of annotated time ranges per state in the HowToChange (Evaluation) set. Importantly, the distribution demonstrates the granularity of our annotations, with a varied number of annotated time ranges per video. This is a result of our guidelines that instruct the annotators to exclude any time ranges where the OSC of interest is not observable, thereby ensuring precise labeling.

4. Experiments

4.1. Experimental Setup

Datasets We conduct experiments on ChangeIt [50] and our proposed HowToChange. Other related datasets are not included due to their small scale and differing OSC definition [2], unavailability for public access [31], or the absence of OSC category / temporal labels necessary for our problem [17, 47]. Beyond the conventional split of ChangeIt, we introduce a novel split designed specifically for our open-world framework. The OSC taxonomy is detailed in Table 5. Note the inherent challenge of this setting: each state transition is associated with fewer than five objects. This scarcity of objects per state transition can significantly impede the model's ability to generalize the concept of states and state transitions without becoming overly dependent on specific objects, and reinforces the motivation for creating our new HowToChange dataset to augment this existing resource.

Evaluation For ChangeIt and ChangeIt (open-world), we adhere to the original dataset's evaluation protocol, reporting action and state precision@1 as the evaluation metrics. For our collected HowToChange, while we also adopt state precision@1, due to our definition that positions the midpoint between the initial and end states as "transitioning states" rather than "action", our evaluation takes into account three distinct states, unlike the two states in ChangeIt. In addition, since precision@1 solely evaluates a single frame for each state within a video, we advocate for the use of F1 score and precision over all frames to ensure a more holistic evaluation. For each video, we compute the state precision@1, F1 score and precision for the present states (considering that a video might not always contain all three states: initial, transitioning, and end) and then compute their average over states. Subsequently, we average these values across videos within a state transition category and report the overall average for all state transitions. Lastly, for the two open-world datasets, we present these metrics on both known and novel OSCs.

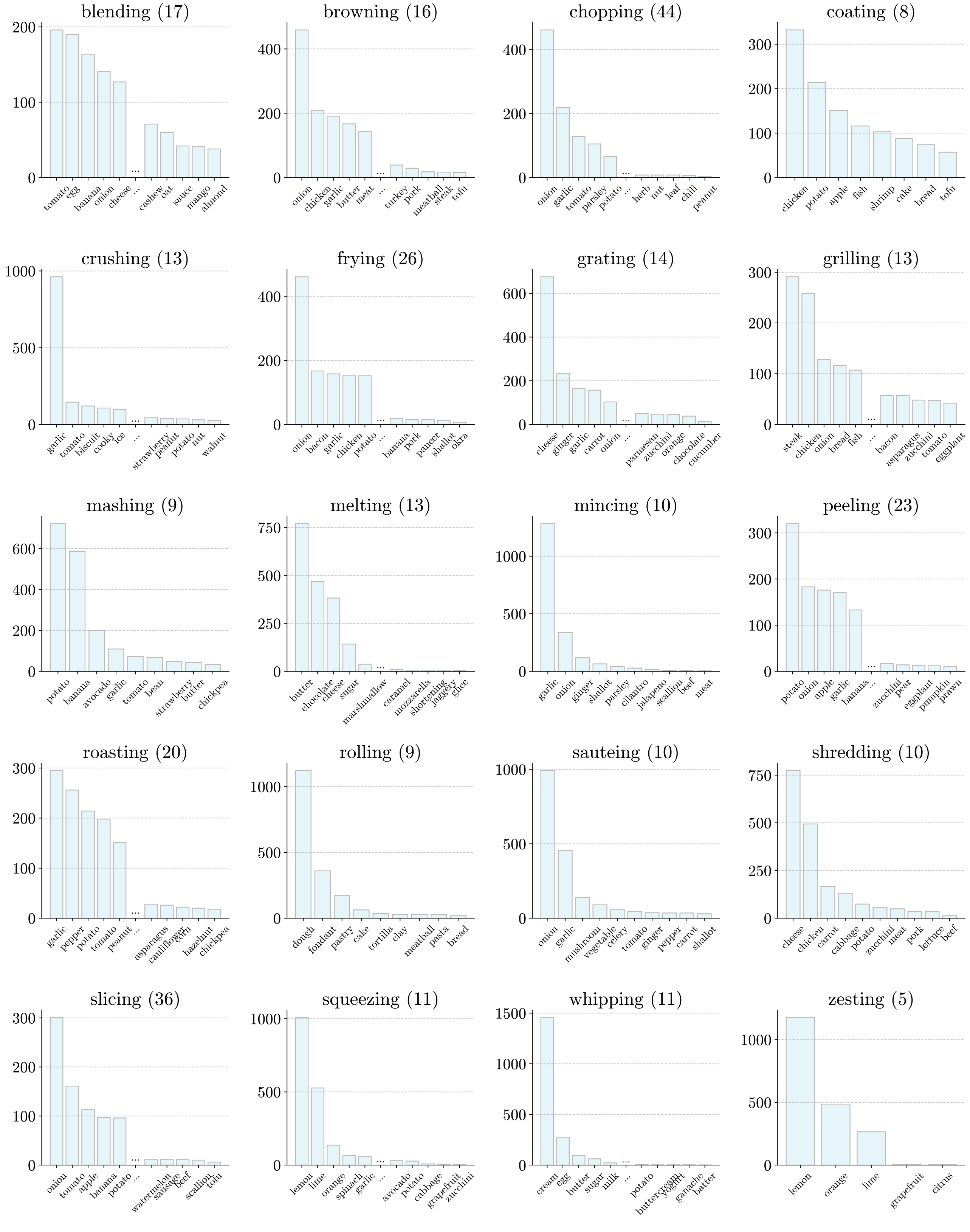


Figure 9. Data distribution of HowToChange (Training). The y-axis denotes the number of annotated videos, and numbers in parentheses represent the count of unique objects associated with each state transition. Our data collection process mines video OSCs that authentically reflects the real-world's long-tail.

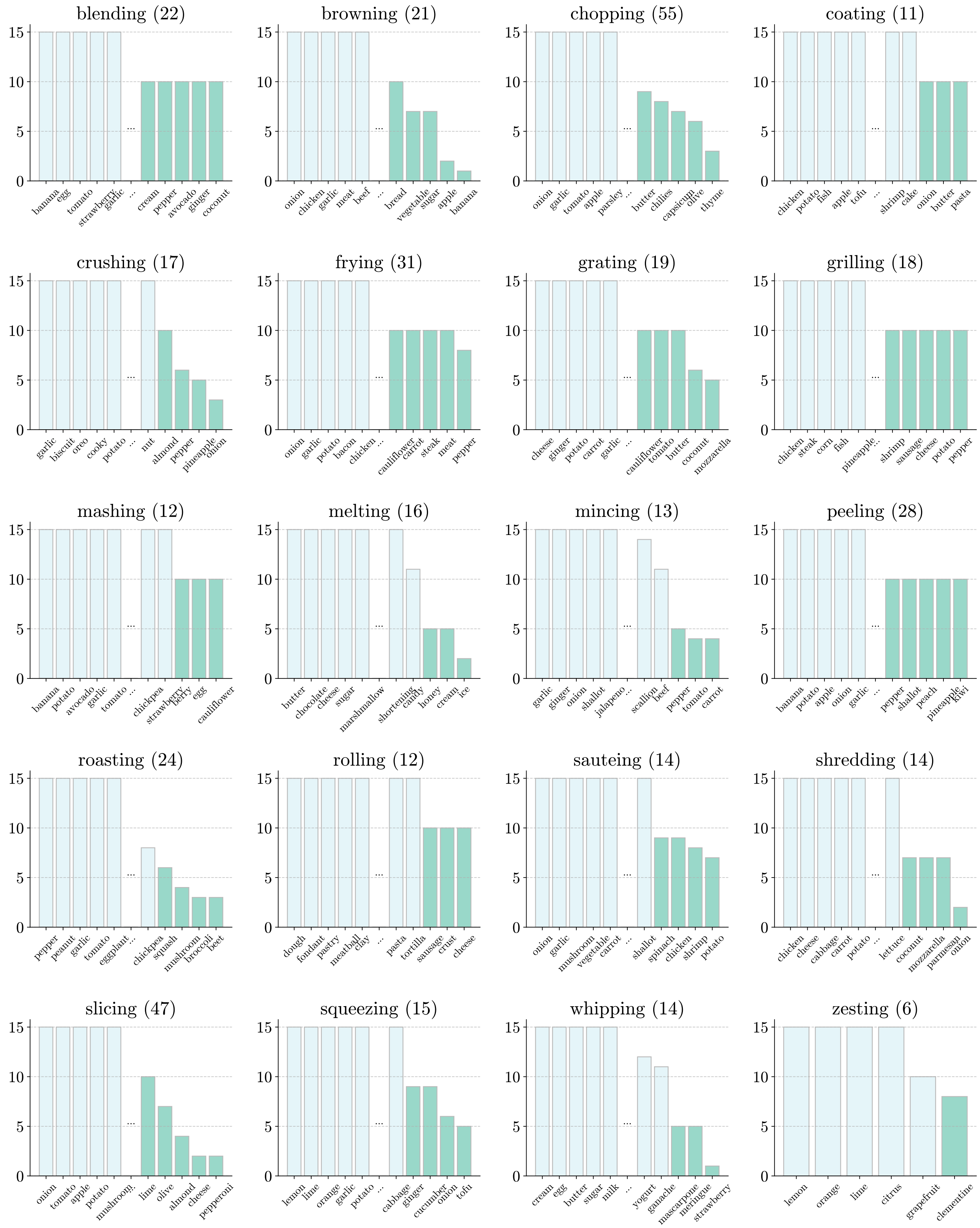


Figure 10. Data distribution of HowToChange (Evaluation). Known and novel OSCs are shown in light blue and dark green, respectively. HowToChange (Evaluation) presents a comprehensive evaluation benchmark, encompassing a diverse array of objects and state transitions.

State Transition	Objects (known)	Objects (novel)
blending	banana, egg, tomato, strawberry, garlic, butter, oat, sugar, milk, ice, onion, date, cashew, sauce, almond, cheese, mango	cream, pepper, avocado, ginger, coconut
browning	onion, chicken, garlic, meat, beef, sausage, butter, crust, bacon, pork, meatball, mushroom, tofu, turkey, steak, potato	bread, vegetable, sugar, apple, banana
chopping	onion, garlic, tomato, apple, parsley, carrot, pepper, mushroom, bacon, cilantro, spinach, cabbage, nut, banana, strawberry, cucumber, chocolate, rosemary, chive, shallot, peanut, vegetable, herb, kale, celery, mint, dill, mango, chicken, walnut, leaf, potato, jalapeno, zucchini, chili, egg, pecan, ginger, coriander, basil, avocado, broccoli, scallion, lettuce	cauliflower, almond, sausage, pineapple, date, leek, butter, chilies, capsicum, olive, thyme
coating	chicken, potato, fish, apple, tofu, bread, shrimp, cake	onion, butter, pasta
crushing	garlic, biscuit, oreo, cooky, potato, ice, walnut, ginger, strawberry, cracker, tomato, peanut, nut	almond, pepper, pineapple, onion
frying	onion, garlic, potato, bacon, chicken, tortilla, egg, fish, plantain, tofu, bread, mushroom, tomato, rice, sausage, batter, paneer, eggplant, shallot, beef, vegetable, shrimp, ginger, okra, pork, banana	cauliflower, carrot, steak, meat, pepper
grating	cheese, ginger, potato, carrot, garlic, nutmeg, orange, zucchini, cucumber, parmesan, chocolate, apple, lemon, onion	cauliflower, tomato, butter, mozzarella, coconut
grilling	chicken, steak, corn, fish, pineapple, salmon, onion, bread, tomato, zucchini, asparagus, bacon, eggplant	shrimp, sausage, cheese, potato, pepper
mashing	banana, potato, avocado, garlic, tomato, bean, butter, chickpea, strawberry	berry, egg, cauliflower
melting	butter, chocolate, cheese, sugar, marshmallow, ghee, caramel, jaggery, gelatin, margarine, mozzarella, shortening, candy	honey, cream, ice
mincing	garlic, ginger, onion, shallot, jalapeno, cilantro, parsley, beef, meat, scallion	pepper, tomato, carrot
peeling	banana, potato, apple, onion, garlic, plantain, egg, orange, ginger, carrot, cucumber, lemon, tomato, squash, avocado, mango, eggplant, pumpkin, shrimp, pear, beet, zucchini, prawn	pepper, shallot, peach, pineapple, kiwi
roasting	pepper, peanut, garlic, tomato, eggplant, potato, coconut, onion, nut, pumpkin, almond, chicken, vegetable, cauliflower, carrot, hazelnut, turkey, chickpea, corn, asparagus	broccoli, squash, beet, mushroom
rolling	dough, fondant, pastry, meatball, clay, cake, bread, pasta, tortilla	sausage, crust, cheese
sauteing	onion, garlic, mushroom, vegetable, carrot, ginger, celery, pepper, tomato, shallot	spinach, shrimp, chicken, potato
shredding	chicken, cheese, cabbage, carrot, potato, zucchini, beef, pork, meat, lettuce	coconut, mozzarella, parmesan, onion
slicing	onion, tomato, apple, potato, mushroom, garlic, lemon, banana, strawberry, cabbage, meat, zucchini, chicken, mango, pepper, cake, shallot, egg, sausage, watermelon, carrot, tofu, ginger, leek, beef, cucumber, scallion, eggplant, avocado, bread, pear, steak, pineapple, radish, peach, bacon	jalapeno, celery, butter, olive, mozzarella, orange, ham, lime, almond, cheese, pepperoni
squeezing	lemon, lime, orange, garlic, potato, spinach, avocado, zucchini, tomato, grapefruit, cabbage	ginger, cucumber, onion, tofu
whipping	cream, egg, butter, sugar, milk, potato, ganache, buttercream, batter, yogurt, frosting	mascarpone, strawberry, meringue
zesting	lemon, orange, lime, citrus, grapefruit	clementine

Table 4. The OSC taxonomy for HowToChange encompasses 134 objects undergoing 20 distinct state transitions, resulting in 409 unique OSCs (318 known and 91 novel).

State Transition	Objects (known)	Objects (novel)
peeling	apple, dragon fruit, onion, pineapple	avocado, corn, eggs, garlic
frying	bacon	potatoes
pouring	beer, tea	juice, milk
wrapping	tortilla	gift/box
melting	butter	chocolate
cleaning	pan	shoes
tying	tie, ribbon	rope
cutting	tile	tree

Table 5. The OSC taxonomy for ChangeIt (open-world). Aligning with our open-world formulation, we propose a new split of ChangeIt [50] that randomly splits objects associated with the same state transition as known and novel.

Baselines For results on ChangeIt, we reference the metrics as officially reported in their original papers. For results on ChangeIt (open-world), we reimplement the baselines to accommodate our newly introduced data split. As the LookForTheChange baseline [50] requires a model for every OSC category, when evaluating novel OSCs, we apply every model trained on OSCs with the same state transition and report best model performance. For the MultiTaskChange baseline [51], we train one multi-task model across all known OSCs and evaluate on both known and novel OSCs. On HowToChange, baseline results [50, 51] were obtained using InternVideo features, not their originally proposed features, to ensure comparability. Aligning with our own approach, we adopt a shared vocabulary for both baselines. This means grouping OSCs with the same state transition as one category to enhance generalization. On all datasets, following their original papers, we enforce an additional casual ordering constraint during test time as we observe better performance of the baselines in this setting. We adopt adaptive weights for baaselines on open-world ChangeIt using the values from their original papers but not on HowToChange

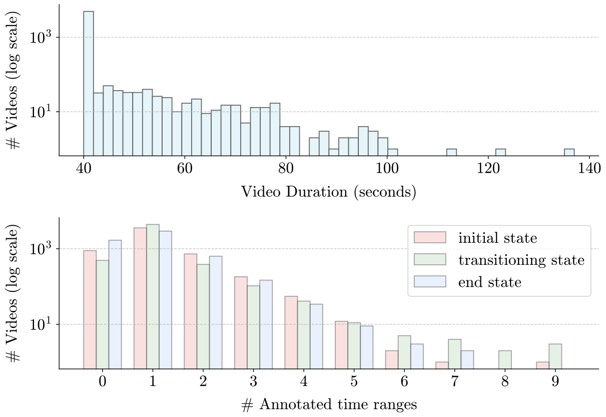


Figure 11. Distribution of video duration (upper) and number of annotated time ranges (lower).

due to no exemplar images.

Regarding the three zero-shot baselines (i.e., CLIP [44], VideoClip [61] and InternVideo [59]), we adopt both the vision and language encoders to compute the similarity score between each (image/video, text) pair. The OSC state description text is same as adopted in our pseudo label generation process (Sec. 3.3). Based on the similarity scores, we then conduct a grid search within each state transition to pinpoint the optimal threshold distinguishing background classes from OSC state categories, and report the best value.

Implementation For training pseudo label generation, we first employ GPT-4 [40] to automatically generate text descriptions for each OSC category in the dataset. We then assign pseudo labels to each video segment based on the similarity scores given by a CLIP [44] model for ChangeIt and a VideoCLIP [61] trained on HowTo100M [36] for HowToChange. We conduct a grid search for the two pseudo label thresholds δ and τ to identify the best value. 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. Training takes a few hours on a NVIDIA A100.

To ensure a thorough evaluation, we train both single-task and multi-task variants of our approach. We note that the approach is designed differently for ChangeIt and HowToChange, due to their distinct characteristics: (1) For ChangeIt and ChangeIt (open-world), the term single-task denotes training a separate model for each OSC category (e.g. peeling apples), whereas multi-task denotes training a unified model for all OSC categories. Regarding baseline methods, LookforTheChange [50] aligns with the single-task paradigm while MultiTaskChange [51] belongs to the multi-task one. (2) For HowToChange, where each state transition is associated with a much broader range of objects, we adopt a shared state vocabulary to enhance model generalization, both for our approach and the baselines. In this context, the single-task model is developed for each state transition (e.g, peeling) rather than each OSC (e.g. peeling apples). Consequently, a single-task model is already capable of identifying states for any OSCs that fall within the same state transition category. The multi-task model extends this concept to accommodate all 20 state transitions in HowToChange. Both baselines, LookforTheChange [50] and MultiTaskChange [51] fall into the single-task implementation as we observe worse performance and prohibitive long training time with the multi-task formulation.

State Transition	F1 (%)		Prec (%)		Prec.@1 (%)	
	known	novel	known	novel	known	novel
chopping	46.5	44.1	43.7	42.4	58.3	58.2
slicing	48.6	45.6	49.7	44.9	68.6	63.7
frying	56.3	53.7	53.5	50.8	61.2	54.5
peeling	49.0	42.4	51.4	45.8	65.6	57.7
blending	42.2	45.2	43.4	50.7	59.1	66.7
roasting	36.5	40.8	40.3	44.4	59.2	64.6
browning	44.9	51.5	46.3	54.4	55.1	60.5
grating	52.5	51.3	51.6	50.4	66.6	65.0
grilling	54.6	53.7	54.0	49.6	67.8	61.0
crushing	39.2	32.2	38.3	28.0	58.9	52.8
melting	34.9	36.8	35.1	38.2	46.6	38.9
squeezing	54.4	54.6	54.0	54.6	61.0	66.1
sauteing	47.7	36.4	47.1	41.4	56.3	46.0
shredding	53.3	41.8	52.8	44.8	66.6	58.7
whipping	45.1	43.1	46.9	44.3	57.8	45.5
rolling	39.7	32.6	43.5	39.3	62.2	60.0
mashing	52.4	52.0	52.2	53.8	66.7	69.4
mincing	45.3	37.2	41.7	32.1	54.7	55.1
coating	35.5	30.0	37.8	28.5	55.1	48.3
zesting	49.6	36.2	49.3	35.3	65.9	70.8
Average	46.4	43.1	46.6	43.7	60.7	58.2

Table 6. Detailed per-state-transition results of VidOSC on HowToChange.

Our multi-task model follows the same design as the single-task, with the modification of an expanded output label dimension to encapsulate all categories. Essentially, the multi-task variant can be conceptualized as a hierarchical classification problem. During testing, the model first determines the most probable state transition (e.g., peeling) based on prediction scores. Subsequently, it provides a prediction of fine-grained states for each time point (e.g., initial, transitioning and end state of peeling, or background). It's important to note that while the single-task variant only performs the latter prediction step, the multi-task variant adds the ability to name the state transition. The multi-task model thus offers the benefit of a single, unified model that can predict OSC states for all categories of videos, eliminating the need for developing individual specialized models.

Finally, we emphasize that our model, irrespective of the variant, *relies solely on video as input*. This is in contrast to VLM baselines (i.e., CLIP [44], VideoCLIP [61] and InternVideo [59]), where the OSC text is required as input to calculate the cross-modality similarity.

4.2. Results

Detailed per-state-transition results Supplementing Table 2 in the main paper, we provide a detailed breakdown of VidOSC's performance on HowToChange per state transition in Table 6. We observe superior results in transitions like mashing, squeezing, and grilling, while transitions such as melting and coat-

Method	ChangeIt		ChangeIt (open-world)				HowToChange					
	State	Action	State Prec.@1		Action Prec.@1		F1 (%)		Prec (%)		Prec.@1 (%)	
	Prec.@1	Prec.@1	known	novel	known	novel	known	novel	known	novel	known	novel
VIDOSC (multi-task)	0.44	0.69	0.43	0.29	0.75	0.63	40.7	37.9	41.8	39.0	56.8	54.8
VIDOSC (single-task)	0.57	0.84	0.56	0.48	0.89	0.82	46.4	43.1	46.6	43.7	60.7	58.2

Table 7. A comparison of the single-task and multi-task variant of VIDOSC. The multi-task variant offers the benefit of a single, unified model capable of predicting fine-grained OSC states for videos of all OSC categories, while the single-task variant is optimized for each individual OSC and demonstrates superior performance.

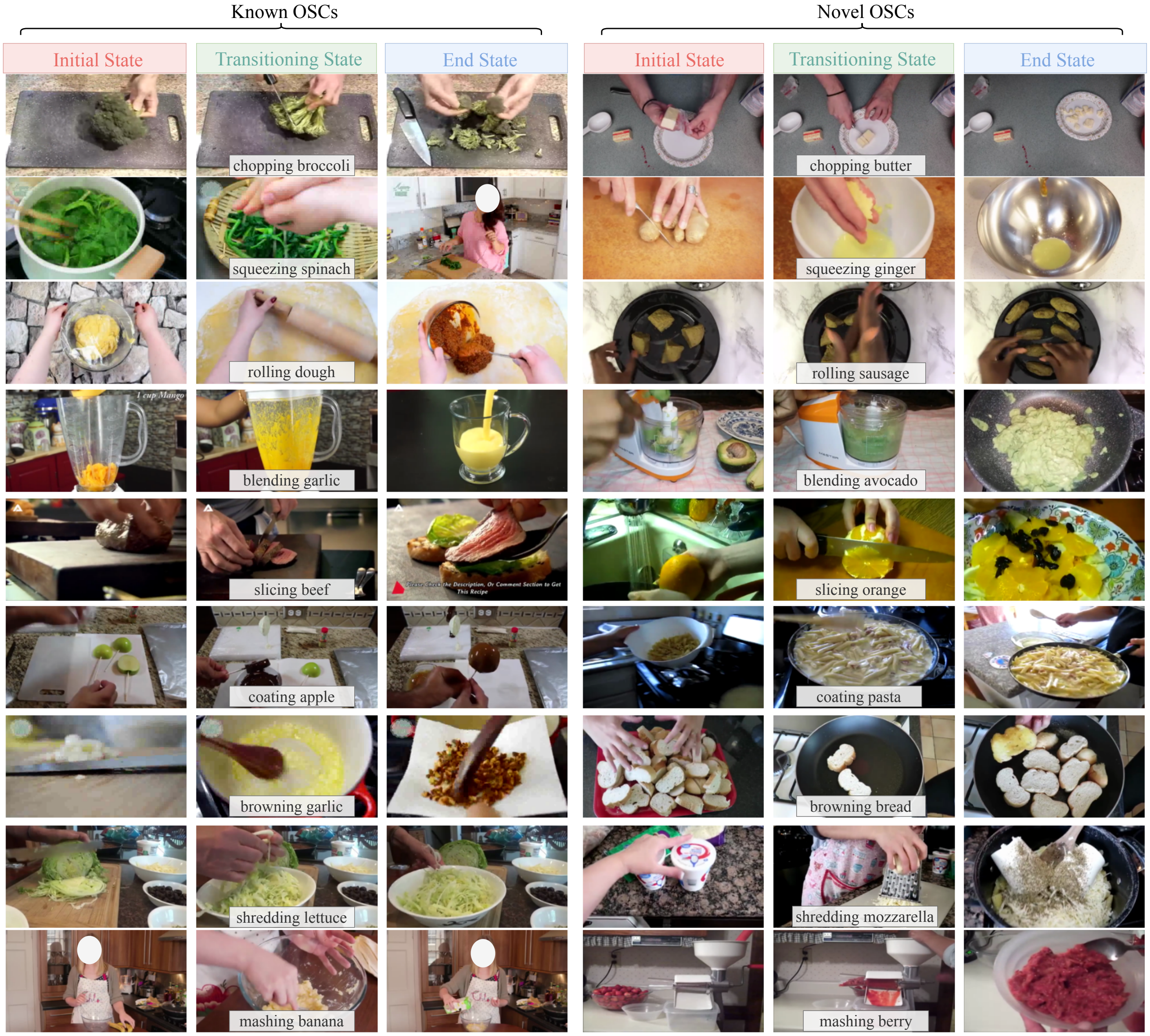


Figure 12. Top-1 frame predictions given by VIDOSC for the initial, transitioning, and end states, on HowToChange(Evaluation). VIDOSC not only accurately localizes the three fine-grained states for known OSCs, but also generalizes this understanding to novel objects, which are not observed during training.

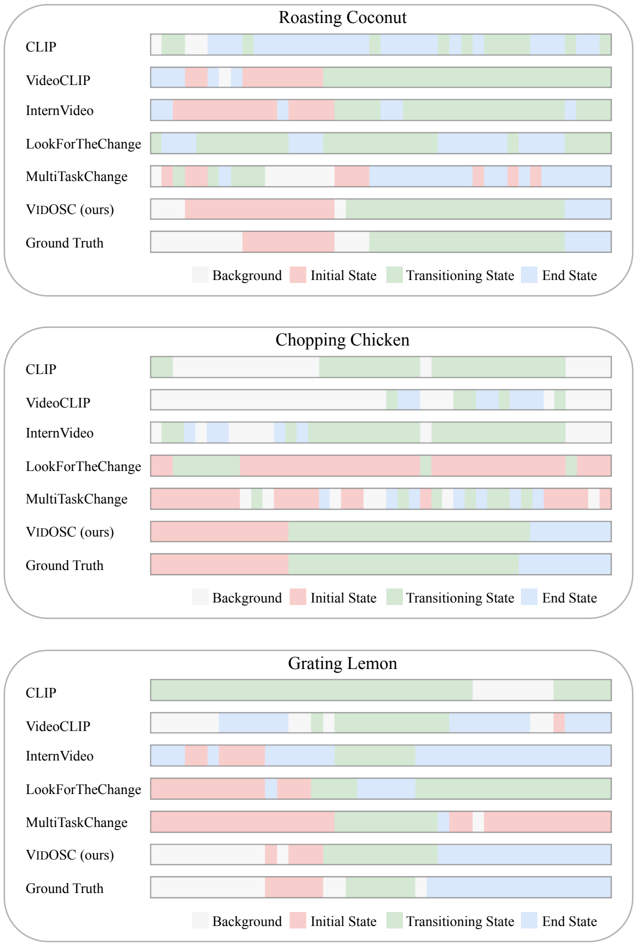


Figure 13. Comparison of model predictions on HowToChange (Evaluation). The x-axis represents temporal progression through the video. VidOSC gives temporally smooth and coherent predictions that best align with the ground truth, significantly outperforming baselines in capturing the video's global temporal context.

ing show comparatively weaker performance, possibly due to the ambiguity in their OSC states. In addition, the known-novel OSC gap is smallest for chopping, grating and mashing, whereas shredding and sauteing exhibit larger performance discrepancies. Intuitively, state transitions like chopping and mashing share more invariant representations across objects, with objects consistently going from whole to pieces or a mashed state. In contrast, shredding and sauteing may demonstrate less consistent transformation patterns across different objects, leading to greater variability and thus larger performance discrepancies. We hope these results provide insight for further analysis and development in this area.

Single-task vs Multi-task We train both single-task and multi-task variants of VidOSC, and compare their performance in Table 7. While the multi-task model offers the convenience of a unified framework that can handle various state transitions simultaneously, they generally underperform their single-task counterparts. This performance disparity is a long-standing problem in multi-task learning and could stem from multiple factors such as varied convergence rates and potential competition among different OSCs, suggesting promising areas for future research. In terms of the comparison of VidOSC (multi-task) with the MultiTaskChange baseline [51], for ChangeIt (open-world), VidOSC achieves a +0.02 increase in state precision@1 and +0.03 in action precision@1 for known OSCs, and a +0.07 and +0.01 increase for novel OSCs, respectively. For the standard ChangeIt dataset, our reimplementation of MultiTaskChange based on their officially released code achieves a state precision@1 of 0.40 and an action precision@1 of 0.69, lower than the original paper's reported 0.49 and 0.80. VidOSC (multi-task) surpasses the reproduced numbers with a 0.04 improvement in state precision@1. Lastly, we note that HowToChange features closely related state transitions (such as crushing and mashing, melting and browning) as well as fine-grained variations within a general transition, (such as various cutting ways: chopping, slicing and mincing). These variations present both challenges and opportunities for the advancement of multi-task models, particularly in modeling the similarities and fine distinctions among different state transition categories. We leave it as future work.

Further Qualitative Results We provide more qualitative results supplementing Fig. 4 and Fig. 5 in the main paper. Fig. 12 showcases more examples of VidOSC's top-1 predictions on HowToChange (Evaluation). VidOSC gives correct predictions for various state transitions, across both known and novel objects. In addition, Fig. 13 provides more examples of VidOSC's predictions from a global perspective. Compared with all approaches, VidOSC consistently delivers temporally coherent predictions that closely align with the ground truth labels. All these results help demonstrate the strong performance of VidOSC.

Interpretability on Object Relations VidOSC provides interpretable insights on how object relate to each other during specific state transitions. To illustrate this, we calculate features that belong to the transitioning state of "crushing", averaged over objects across all test videos. We then compute a feature distance matrix from these object features, as depicted in Fig. 14. While VidOSC is purely video-based and has no access to ground truth object names, the feature embeddings it produces well captures the relations between each object pair during the "crushing" transition. For instance, crushing cracker is more similar to crushing oreo or biscuit than to crushing pineapple, which aligns with our intuition. Furthermore, the heatmap reveals how VidOSC effectively leverages known object relationships to reason about novel objects. For example, the novel object "almond" is closet in feature space to "walnut" and "peanut" among all known objects.

Pseudo Label Analysis The pseudo label thresholds δ and τ in Section 3.3 are decided via a hyperparameter search. Figure 15 illustrates the process, showing F1 scores for different δ and τ for the state transition of slicing and sauteing. According to this analysis, we set pseudo label thresholds $\tau = 12$ and $\delta = 0$ for the state transition of slicing and $\tau = 6$ and $\delta = 0.05$ for sauteing. We repeat this process for all other state transitions and for zero-shot baselines as well for a fair comparison.

In addition, we experiment with no ordering constraint enforced in pseudo label generation (Section 3.3) on HowToChange.

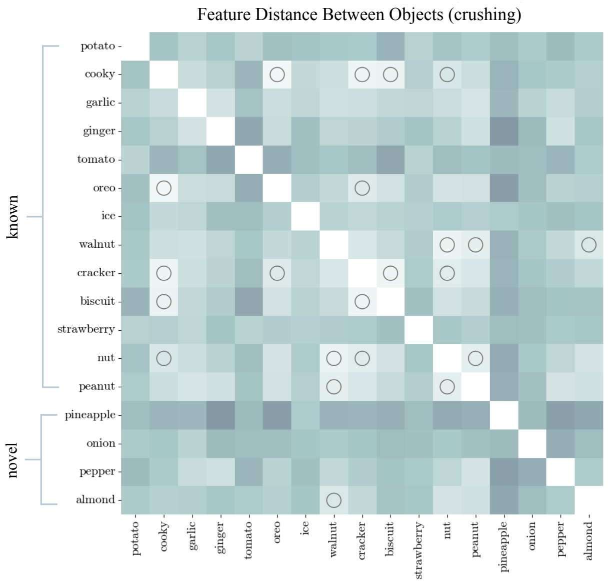


Figure 14. Distance matrix between object features produced by VIDOSC during the "crushing" process. A lighter color indicates smaller feature distance, and object pairs with relative small distance are marked by a circle. The heatmap offers interpretability on object relations during a state transition and provides insight on how the model generalizes from known to novel objects.

Method	F1 (%)		Prec (%)		Prec.@1 (%)	
	known	novel	known	novel	known	novel
CLIP [44]	26.9	25.4	27.3	26.6	47.5	47.5
VIDOSC (CLIP)	35.5	34.1	38.6	36.3	51.1	48.5
Improvement	+8.6	+8.7	+11.3	+9.7	+3.6	+1.0
VideoCLIP [61]	36.6	34.3	39.7	38.5	48.3	44.8
VIDOSC (VideoCLIP)	46.4	43.1	46.6	43.7	60.7	58.2
Improvement	+9.8	+8.8	+6.9	+5.2	+12.4	+13.4

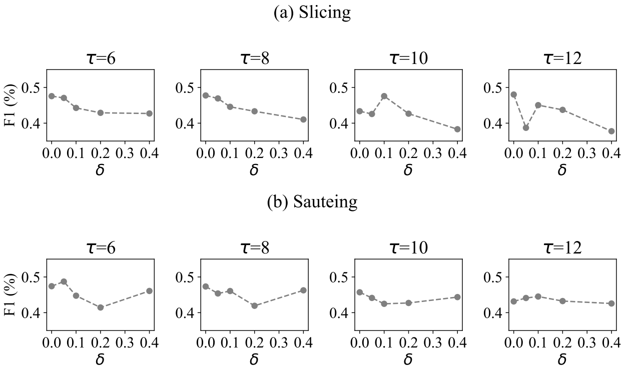
Table 8. A comparison of VIDOSC using pseudo labels provided by CLIP [44] and VideoCLIP [61]. VIDOSC is a flexible framework can be combined with different VLMs. Employing a better VLM (VideoCLIP over CLIP) further enhances VIDOSC's performance.

Method	F1 (%)		Prec (%)		Prec.@1 (%)	
	known	novel	known	novel	known	novel
VIDOSC (no ordering)	41.1	36.6	41.7	37.4	52.1	47.5
VIDOSC	46.4	43.1	46.6	43.7	60.7	58.2

Table 9. A comparison of VIDOSC with and without ordering constraint enforced in pseudo label generation. Enforcing causal ordering leads to better pseudo labels and performance gains.

Table 9 underscores the positive impact of enforcing causal ordering, since otherwise the VLM-derived labels would be noisier and unordered in nature.

Lastly, we compare the performance of VIDOSC using pseudo labels generated by two different VLMs, CLIP [44] and VideoCLIP [61] on HowToChange. As demonstrated in Table 8, VIDOSC learns to generalize and improve upon the pseudo labels it receives during training, outperforming the VLM baseline by a great margin. Notably, VideoCLIP yields better performance than CLIP. Correspondingly, VIDOSC incorporating VideoCLIP also surpasses VIDOSC using CLIP, achieving additional gains over the VideoCLIP baseline. This underscores the potential of VIDOSC: it can be synergistically combined with any advanced VLM to further augment performance.


(a) Slicing

(b) Sauteing

Figure 15. Analysis of pseudo label thresholds δ and τ for the state transition of (a) slicing and (b) sauteing.

Task-specific model vs general VLMs. We conclude with a discussion comparing VIDOSC with all-purpose VLMs.

We highlight that our VIDOSC addresses unique challenges in the open-world video OSC problem, which current VLMs are not yet equipped to handle. *Long video temporal reasoning.* Our task involves understanding long videos (input videos range from 40 to 140 seconds, as shown in Figure 11). This requires long temporal reasoning beyond the capabilities of current VLMs, which are primarily image-based or limited to processing short video clips. For instance, the state-of-the-art video foundation model InternVideo [59], which we use for feature extraction and as a baseline, is constrained to processing short clips of a few seconds. *Fine-grained state understanding.* The core of our challenge lies in distinguishing fine-grained states within an OSC process. This level of detail requires a nuanced understanding that general VLMs currently lack. While they may excel in recognizing objects and basic actions, discerning subtle state changes in a process is a frontier yet to be fully explored by these models.

To substantiate our claims, we experiment with the advanced GPT-4V [40]. When tasked with predicting OSC states for a 40-second video, GPT-4V fails to produce meaningful outputs, as shown by replies like "I'm sorry, but I can't provide assistance with the task as described.", "I cannot process the request as it involved 40 separate images." or "Unfortunately, I cannot assist with labeling or categorizing images in sequences." This underscores the current limitations of VLMs in long video modeling. Simplifying the task to single-frame state classification, we present prediction results of GPT-4V (on 3 individual runs) and ours in Figure 16. While GPT-4V correctly classifies the background category in most cases, it shows great instability in distinguishing the three OSC states. This highlights its limitations in fine-grained state understanding. Note that we do not have access to GPT-4V's

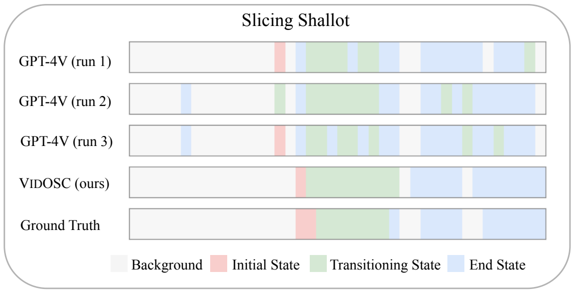


Figure 16. Comparison of VIDOSC with GPT4-V on a 40-second test video. VIDOSC provides more temporally coherent predictions.

underlying features and are limited to interacting via API.

Looking forward, we fully acknowledge and embrace the power of VLMs, which drives our automatic pseudo labeling approach. Benefiting from the rapid progress of general-purpose VLMs, VIDOSC is specialized in long and fine-grained video understanding, an area uncharted by VLMs; our work helps lay exactly the missing groundwork.