EPIC-KITCHENS VISOR Benchmark
VIdeo Segmentations and Object Relations – Appendix

This appendix contains additional details about the VISOR dataset.

§A Appendix - Societal Impact and Resources Used describes the societal impact of this dataset, the resources used, and the availability of the dataset. This is also expanded upon in the datasheet in §K.

§B Appendix - Entities, Frames, and Subsequences (Main §2.1) describes the selection of entities to be annotated, frames on which annotations are made, as well as subsequences. This is the first part of our annotation process and sets up the pixel-wise annotations.

§C Appendix - Annotation rules and annotator training (Main §2.2) / Annotator Training and Rules describes the training of the annotators the TORAS annotation suite.

§D Appendix – Tooling: The TORonto Annotation Suite (Main §2.2) describes the TORonto Annotation Suite (TORAS) that was used for annotating the pixel labels.

§E Appendix – Correction (Main §2.2) / Correction describes the corrections done to these annotations.

§F Appendix - VISOR Object Relations and Entities (Main §2.3) describes the annotation of the relationships between the segments. This is the last part of our annotation.

§G Appendix - Dense Annotations (Main §2.4) describes the dense annotations that we provide and how they were obtained.

§H Appendix - VOS Benchmark Details (Main §4.1) describes the Video Object Segmentation (VOS) benchmark, including data preparation (§H.1), metrics (§H.2), baselines (§H.3), and additional results (§H.4).

§I Appendix - HOS Benchmark Details (Main §4.2) describes the Hand Object Segmentation (HOS) benchmark, including data preparation (§I.1), metrics (§I.2), baselines (§I.3), and additional results (§I.4).

§J Appendix - WDTCF Benchmark Details (Main §4.3) describes the *Where Did This Come From?* benchmark, including data preparation (§J.1), metrics (§J.2), baselines (§J.3), and additional results (§J.4).

§K Appendix - EPIC-KITCHENS VISOR - Datasheet for Dataset describes the datasheet for the VISOR dataset.

Supplemental References

Implementation details in the appendix refers to two other items that were not reference in the main paper. We have put these references here:

[A] Georgia Gkioxari, Ross Girshick, Piotr Dollar, and Kaiming He. Detecting and recognizing human-object interactions. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[B] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019

A Appendix - Societal Impact and Resources Used

Dataset Bias and Societal Impact. While the EPIC-KITCHENS videos were collected in 4 countries by participants from 10 nationalities, it is in no way representative of all kitchen-based activities globally, or even within the recorded countries. Models trained on this dataset are thus expected to be exploratory, for research and investigation purposes.

We hope fine-grained understanding of hand-object interactions can contribute positively to assistive technologies, for all individuals alike including in industrial settings. Approaches for imitation learning are expected to benefit from VISOR. We hope future models will replace mundane and dangerous tasks.

Computational Resources Used.

Estimating the precise computational resources used over the course of a 22 month project is challenging. However, we give a sense of the computational requirements and briefly report information about the computational resources used by each component.

TORAS. TORAS uses a server with 2 GPUs to run the interactive segmentation interface, one per model.

VOS Baseline. For both traning and inference, we used a single Tesla V100 GPU. Training took 4 days on VISOR.

Dense Annotations. We use the model from VISOR baseline to extract dense annotations, and calculate their scores. This took 7 days using 9 Tesla V100 GPUs. Our code is designed to use 3 GPUs but we ran multiple parallel instances.

HOS Baseline. Training the PointRend model as described took 2 days on 2 A40 GPUs.

WDTCF Baseline. Most of the computational cost of WDTCF is in our PointRend model. Training the PointRend model as described took 2 days on 2 A40 GPUs.

Annotation Done. We estimate that the total number of hours to annotate the data is around 25,000 hours - including human annotations, manual checks and corrections. It is challenging to estimate correctly the resources used as the server is used for multiple projects. The bulk of the annotation work was done in the pixel labeling. Freelance annotators who did this were paid hourly, and self-reported their hours via the Upwork platform. Hourly rate varied by experience and submitted proposal from $6-$9/hour, with the higher rate reserved for most experienced annotators who were also in charge of QA checks. We believe annotators were fairly compensated, working directly with the freelance annotators. Small amounts of the data were annotated via crowdsourcing services. We report the cost spent in those sections.

Storage and Availability. As with the EPIC-KITCHENS-100 videos, the VISOR annotations is permanently available with a unique DOI from the University of Bristol Data-Bris storage facilities at: https://doi.org/10.5523/bris.2v6cgv1x04ol22qp9rm9x2j6a7. The data management policy for Data-Bris is available from http://www.bristol.ac.uk/staff/researchers/data/writing-a-data-management-plan/. The University of Bristol is committed to storing, backing-up and maintaining the dataset for 20 years, in-line with UKRI funding requirements.

B Appendix - Entities, Frames, and Subsequences (Main §2.1)

This stage identifies *what* will be annotated in terms of entities (i.e., names of the objects) and frames (i.e., which frames to annotate). The goal of this stage is to generate an **overcomplete** set of entities which annotators will later select from as well as identify a list of frames that can be accurately annotated. We describe the preparation of entities (§B.1), selection of frames (§B.2), and examples of our subsequences (§B.3).

B.1 Entity Preparation

Entity preparation is the cornerstone of the entire project. Annotators refer to the entities list to provide manual pixel-level segmentation. Therefore, the quality of the entity list directly determines the quality of the proposed dataset. In order to ensure quality, we first automatically extract potential active entities from narration, then task workers on Amazon Mechanical Turk with creating a list of active entities. Pixel-label annotators can choose to ignore an entity if absent, occluded, highly-blurred or incorrect altogether.

Entity Extraction. Entity candidates consist of a main-object list and three additional-object lists. A main-object list contains nouns appearing in the narration and additional-object lists include the commonly used tools. We then detail both lists.

Main list. We extract entity candidates per short 'action' clip in EPIC-KITCHENS-100 [11] to form the main list. This dataset provides not only the ground-truth of verb and noun for action recognition, but also the narration for each clip. Naturally, the objects in the narration are active. Hence, entities for one clip are extracted from the narration.

Adjacent Clips. However, these are not enough as we found that most active objects are not mentioned in the narration, (for instance 'peel potato'). In this case, 'peeler' is the active object but it does not appear in the narration. Thanks to the density of the annotation of EPIC-KITCHENS-100 [11], some missing objects are likely to be found in previous or latter actions, for example 'take peeler' before 'peel potato'. Therefore, the candidate entities for one clip are formed by merging the entities from the current clip, the previous four clips, and the ensuing two clips. Moreover, dense annotation of EPIC-KITCHENS-100 [11] might cause overlapping clips, meaning some frames belong to both clips. This overlap might lead to replicated work for annotators. To solve this issue, we generate the candidate entities for non-overlapping clips.

Common Tools. Although looking at adjacent portions of the video and narration can help us find missing active objects, it still has the limitation of missing objects mentioned in far away clips or even not mentioned at all. For example, in 'cut celery', the 'chopping board' never appears in narrations but it indeed is an active object. To tackle this problem, we propose three additional-object lists which include the high-frequency objects in terms of cutlery, utensils, as well as furniture. They contain ['fork', 'knife', 'spatula', 'spoon', 'other cutlery'], ['chopping board', 'bowl', 'plate', 'cup', 'glass', 'pan', 'pot'], and ['cupboard', 'drawer', 'tap', 'drainer', 'hob', 'bin/garbage can/recycling bin', 'fridge', 'oven', 'sink'], respectively.

Americanisms. Finally, 34 British entity names are translated to American alternatives. Thus, we provide a main-object list and three additional-object lists to workers in Amazon Mechanical Turk to select the proper active entities.

Entity Selection. The proposed method for entity extraction brings some irrelevant entities. We design an interface for workers to select the proper active entities. The interface, shown in Fig. 10, is composed of six parts. First of all, it illustrates instructions to guide workers to complete the work step by step. Below the instructions, there is a clip introducing the action which is highlighted in red on the right. The "Chosen List" shows the ticked selection and the "reset" button clears all selections. Next, the interface shows our prepared entities. Workers then select the entities from main-object and additional-object lists step by step while referring to the clip. All the selections are collected with the "Submit" button.

Additionally, in order to improve the quality, we design some functions for the interface. Specifically, (1) some entities selected in the previous clip will be automatically ticked in the current clip. We notice that a clip is very likely to inherit objects from the previous clip due to the continuity of the video and dense annotations of EPIC-KITCHENS-100 [11], e.g., "take bowl" and then "wash bowl". If the ticked entities are irrelevant to the action, the "reset" button can help to clear all selections and

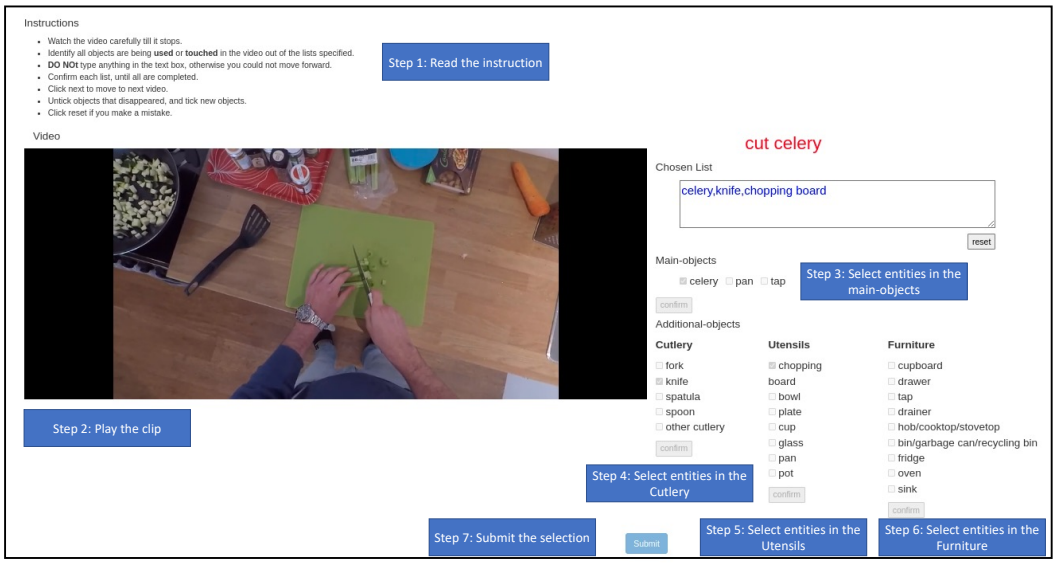


Figure 10: **AMT interface for entities selection.** This is used to identify *which* objects will be given pixel annotations.

Table 6: **Frame rate variations statistics**. Variable frame rate used in the majority of the videos. We compare these to fixed size sampling at 2fps, collected for 4 videos.

	avg. rate	videos	seq	images	masks	%hands	max frames/seq
Variable frame rate	0.9fps	172	7552	44,777	239,357	30.0%	6
Fixed frame rate	2fps	7	308	5,952	32,227	32.4%	186

workers are able to restart selecting entities. (2) The interface requires workers to watch the clip first before choosing, otherwise they cannot proceed. Also, 3) it asks the worker to select candidates one by one in order by clicking on different "confirm" options. (4) If an entity appears in the main-object list, it will be removed in additional lists.

We use Amazon Mechanical Turk to complete entities selection. At the very beginning, we set five different workers to complete one sequence composed of 16 consecutive clips. The reward per sequence is \$0.18. We merge all selections using majority decision. Later, we decreased the number of workers to two as we learnt from current examples and introduced our correction stage described next. When a word is chosen by 4/5 or 2/2 Amazon Mechanical Turk workers, we consider it as an active entity. Additionally, we propose two rules for merging selections. One is verb-entity, which introduces new entities based on the verb used, e.g., 'wash' in 'sink', 'stir' on 'hob'. Another one is entity-entity, e.g., 'tap' with 'water' in 'sink' for instance. By combining the selections from different workers, we acquire our over-complete list.

Entity preparation matters. To compare the difference between before and after entity preparation, we visualise the changes in the number of entity classes in Fig. 11. The difference between orange and blue entities shows the additional entities found through entity preparation. Additionally, it substantially increases the number of entity classes that appears before while bringing right and left hands. Note that the occurrence of hands is reasonable given our egocentric videos. The number of "sink" entities is increased the most due to the prevalence of the washing actions.

B.2 Frame Extraction

For the majority of videos in VISOR, we use an approach of variable frame rate for frame selection. We sample 6 frames per subsequence, where a subsequence is composed of 3 nonoverlapping actions. At the end of the video, when only 1 or 2 actions remain, we sample less frames accordingly.

For only 7 videos in VISOR, we attempt a fixed frame rate in line with other datasets. Table 6 shows the statistics of each variation.

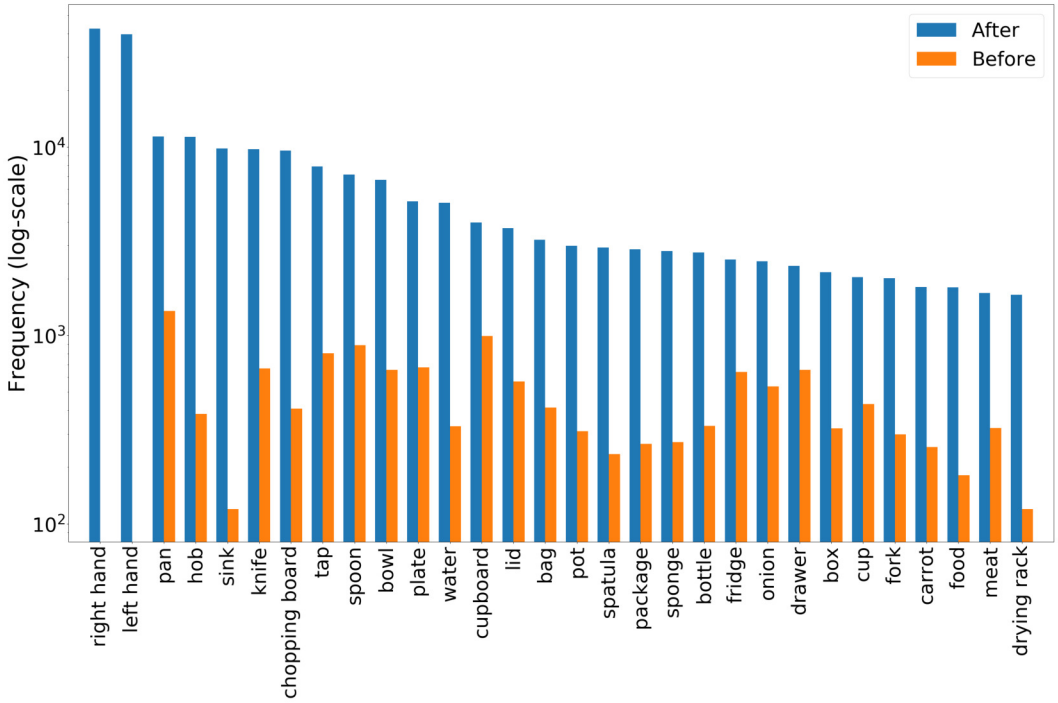


Figure 11: **Changes in the number of entity classes before and after entity preparation.** Entity classes are ordered by frequency of occurrence. Our entity preparation is critical.

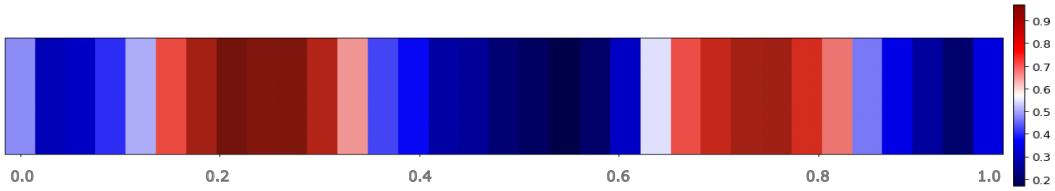


Figure 12: **Distribution of the frames per action**. we considered frames within actions only.

Why use a variable frame rate? A variable frame rate is useful to: (1) concentrate the sampled frames to be within actions (rather than between actions); this concentration leads to more non-hand objects as shown in Table 6. Concentrating in this way is helpful in all of our benchmarks, as 3.7% of the frames are between actions in the variable frame rate scheme whereas 10.2% are between actions when using a fixed frame rate. (2) Save annotation time to include more videos. (3) potentially add to the sequence difficulty of video-based benchmarks such as semi-supervised VOS as there is no count limit between the annotated frames.

Why also include a fixed frame rate? A fixed frame rate is the standard in other benchmarks. We decided to collect videos this way so as to enable researchers to appreciate the difference between the two regimes. This also allows denser evaluations, and so 4 of our 7 fixed frame rate videos are left for the Test set. We release 3 videos annotated at 2fps in Train/Val.

How are the 6 frames per subsequence sampled in the variable frame rate? We sample 2 frames per action at 25% and 75% completion of the sequence, then we apply random frame shift of $\pm 10\%$. Our choice of 2 frames per action allows annotating the transformation of objects within the action when present. Next, we apply a Laplacian filter with a window of 20 frames (± 10) to select the frame with the lowest motion blur (lowest variance). Finally, to avoid frames with similar appearance we check if any 2 frames are closer than 25 frames (~ 0.5 seconds). If so we re-sample one of them to be in the middle of the farthest 2 frames in the subsequence. This enriches the subsequence's temporal information.

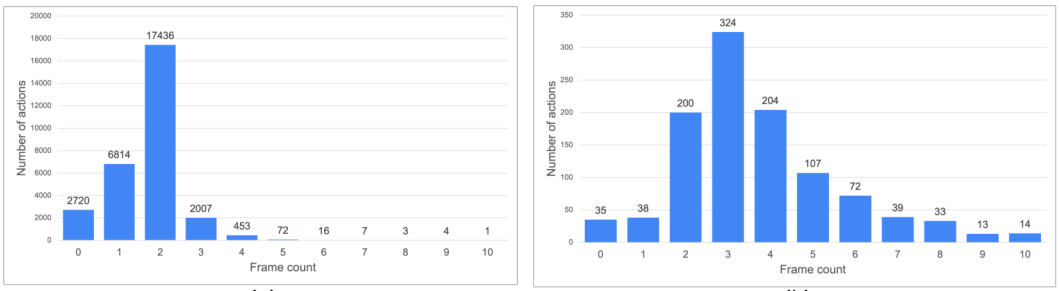


Figure 13: **Distribution of the actions by each frame count.** (a) variable frame rate; (b) fixed frame rate. Both distributions show the first 10 frame counts.

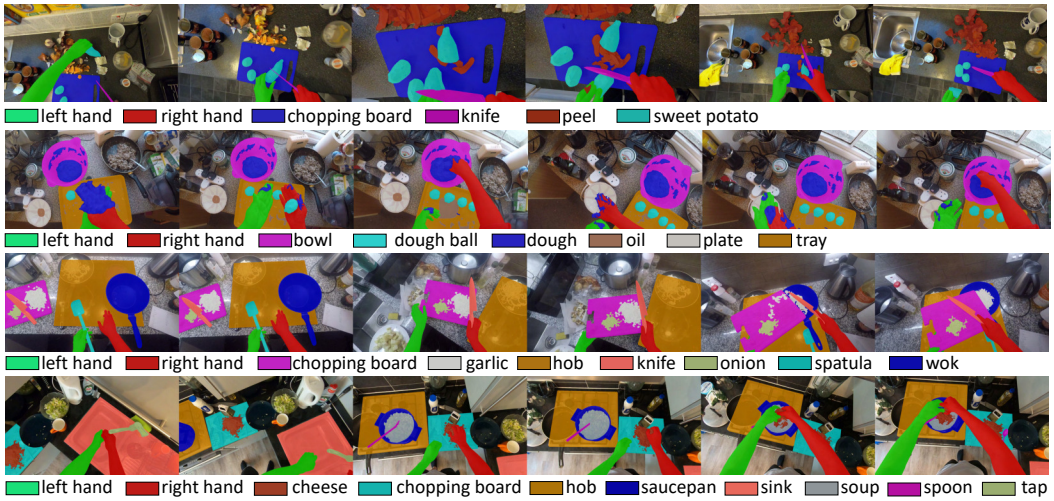


Figure 14: **Four subsequences from different videos.** In each row, we plot overlays from six consecutively annotated images, with temporally consistent segmentations (legend). Actions are 'peel sweet potato', 'make dough ball', 'pour garlic' and 'put cheese', respectively.

How are the frames distributed in the actions? Fig. 12 shows the actions' frame distribution of the whole dataset, the selected frames are concentrated around 25% and 75% of the way through the action's length.

How many frames do most actions have? Fig. 13 shows the distribution of the number of frame counts per action, comparing the variable frame rate and fixed frame rate strategies. Most actions in the variable frame regime have 1-2 sampled frames. There are some actions (2720) without any annotated frames. These are typically short actions that overlap with other actions. As our subsequence focuses on non-overlapping actions, these overlapping actions might not have frames sampled within their boundaries. For the fixed frame rate, most actions have 2-4 sampled frames which is significantly higher than the variable frame rate. There is also a long tail of many frames that have many, many frames (e..g, 171 actions have six or more frames). This means that much of the annotation budget is spent on annotating the same action repeatedly.

B.3 Subsequence Examples

In Fig 14, we show 4 subsequences from various videos. As noted in the main paper, we use the term 'subsequence' to refer to the 6 frames from 3 consecutive non-overlapping action labels in EPIC-KITCHENS. These have a consistent set of entities throughout. As shown in the figure, entities are temporally consistent over these short-term subsequences, demonstrated by consistent legend under each subsequence.

C Appendix - Annotation rules and annotator training (Main §2.2) / Annotator Training and Rules

Once entities and frames have been identified, we annotate the frames with the labels. A key step is the training of annotators. We now describe their recruitment (§C.1), training (§C.2), and the rules that they followed (§C.3).

C.1 Annotator Recruiting

We hired annotators via Upwork due to its convenient system for communicating with annotators. We started with a large pool of freelancers, with whom we shared a document explaining the AI tool. The freelancers then annotated a common set of images and we encouraged them to use the AI tool. We then selected a smaller subset based on annotation speed and quality, and quick adaptation to use of the tool.

C.2 Training Material

To train annotators, we designed an onboarding website (with salient pages shown in Fig. 15). The onboarding website functions like a tutorial. The main pages of the tutorial are listed next:

1. an overview of the project and the role of annotators, so trainees can get familiar with the task.
2. how to use the AI annotation tool, specifically the functionalities of the various buttons in the interface.
3. step-by-step annotation of a single sequence, where every action taken across the course of the video is explained in detail.
4. a breakdown of the annotation rules, highlighting example images where each of the rules applies, and how one would segment objects in relevant situations.
5. an FAQ, with general questions as well as an explanation for some confusing corner cases encountered with the tool.
6. a quiz to test annotators for understanding.
7. some tricky/difficult annotation examples, and how these should be segmented.
8. a glossary of all the object classes annotators need to be familiar with, as well as examples for each class. We opted to include the glossary to avoid any guessing from non-native speakers on what a 'spatula' or 'sieve' are. The glossary is populated via a Google Image search, where for every object we request to be labelled in the dataset, we display the top-3 images that a Google search for that object returns.

C.3 Rules

The rules are designed to maximise consistency and minimise errors when using the AI automatic annotation tool. Workers follow rules about *how* to segment to ensure temporal consistency within a video and *what* to segment to ensure consistency across the dataset.

How to Segment. As segmentations are only provided for certain in-contact objects and objects that occlude them, the first instruction is to consider the six frames that make up a subsequence, and the objects requested for segmentation, in order to identify which objects should be segmented. Next, the user is encouraged to proceed frame by frame. Within a frame, they segment objects visible in the frame if present. They also mark objects that are listed but are invisible in that frame. After segmentations for a frame are complete, they should press the submit button.

What to Segment. After this overview of "how to segment", we next explain "what to segment" with Fig. 16. We explain each of the rules in the caption as these correspond to a single example in the figure.

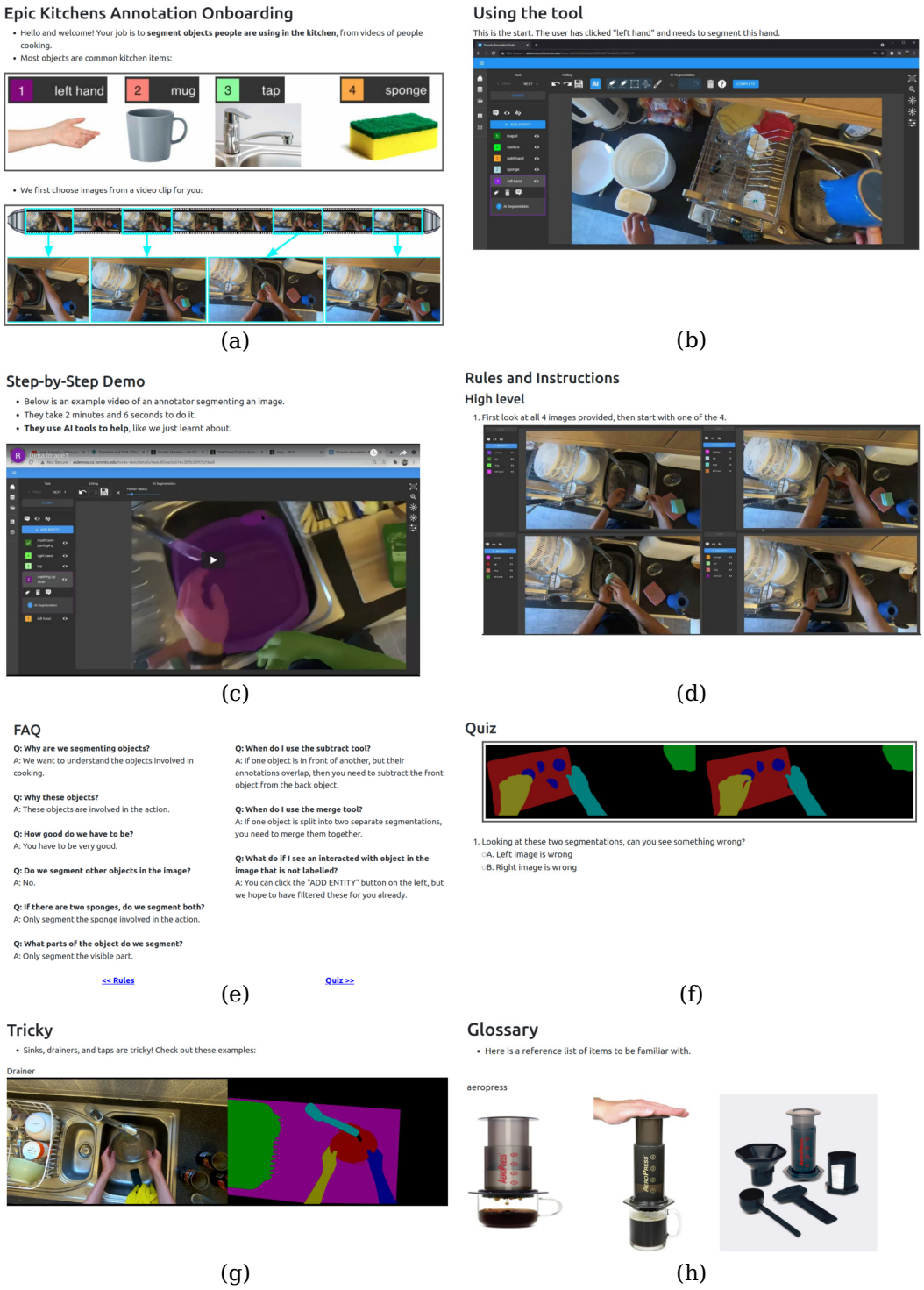


Figure 15: **A page-by-page breakdown of the onboarding process.** (a) A front-page that introduces the project. (b) An overview on using the tool. (c) A step-by-step fine-grained walkthrough explaining everything that happens in the example annotation video. (d) The complete rule handbook with examples and instructions per rule. (e) An FAQ for tricky situations and general information on the project. (f) A quiz to test for learning. (g) Selected challenging situations that arise when annotating. (h) A glossary of items in the dataset, including example images for each class.

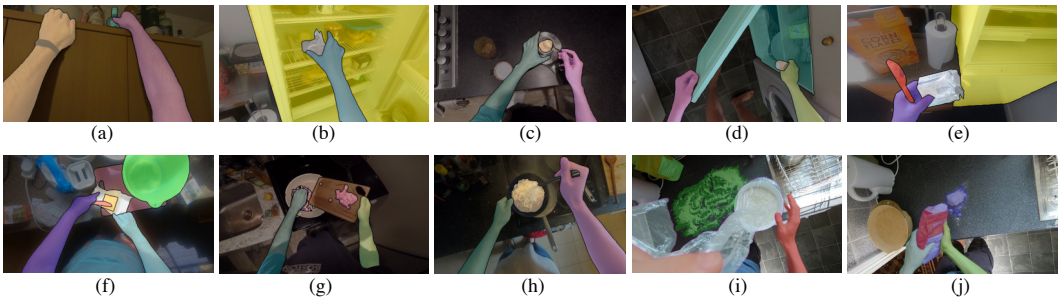


Figure 16: **Demonstrations of rules for hands and active objects segmentation.**
- **(a)** For left and right hands, segmentations include all visible parts of the hand and arms. For consistency, we requested that annotators include watches or wristbands on both hands.
- **(b)** For situations where a container should be segmented but objects internal to the container are not requested for segmentation, we request they segment the entire container, e.g., all of a fridge. If an internal object is requested and visible, the user should separately segment the visible parts of both, e.g., the butter package and the fridge in (b).
- The most tricky cases are to segment the contained objects. Specifically, **(c)** if the object is contained within another (but still partially or fully visible), we ask the user to only segment the visible parts of the contained object and ignore the container. This allows us to segment both the 'paste' and the 'jar' separately in this example.
- Instead, **(d)** if an object is fully contained within another (and thus invisible) e.g., 'salt' inside a bottle, we ask the user to segment the whole container, i.e. the 'bottle' as the 'salt' in this case. In this case, when the user is retrieving the 'salt' from the cupboard, they are in fact segmenting the salt bottle which has salt inside.
- **(e)** If objects are fully contained but invisible, and packaging is requested, we ask the user to segment only the packaging, e.g., the butter package.
- Next, **(f)** if partial occlusions occur, we instruct the user to only separate the visible parts of the occluded objects. For example, the knife is above the butter which is in the package.
- **(g)** For objects with several small pieces, e.g., the pieces of carrots, we instruct users to segment each piece individually and merge the segmentations into one.
- However, **(h)** we mention that if objects are truly tiny and overlapping, e.g., noodles, they can have their internal borders ignored. - **(i)** If a transparent object, e.g., packaging, can be segmented, we instruct users to segment it.
- Finally, **(j)** we tell users that they can ignore this rule if the packaging is extremely transparent, thus implausible to segment, and the contained object is much more visible to segment accurately.

D Appendix – Tooling: The TORonto Annotation Suite (Main §2.2)

Annotating our dataset is made substantially easier by the use of an AI-assisted annotation tool, the TORonto Annotation Suite (TORAS) [20]. This section summarises the features and corresponding interfaces that were used in annotating the segmentation masks for EPIC-KITCHENS VISOR. The TORAS interface has been assessed and approved by the University of Toronto ethics review committee.

The TORAS interface was used to annotate segmentation masks for a fixed list of objects per image derived from entity identification. Later, the interface was used by annotators to correct segmentations and add new objects by incorporating comments collected after a round of corrections. Specifically, we discuss the segmentation tools, including AI enabled tools made available to annotators (§D.1), project management features (§D.2), report annotator agreement statistics (§D.3) and discuss annotation efficiency(§D.4).

D.1 Segmentation Tools

In this section, we describe the TORAS tools used to create and edit segmentation masks. Note that we always use polygons with floating point coordinates as the base representation to ensure high precision segmentations can be created. We also note that annotators have the ability to not use the AI through a toggle button. On beginning annotation of a task, annotators see on the left a list of objects to annotate, pre-populated using entity identification annotation outputs. Annotators can select any object at any time and continue annotating it. The annotators can choose to lock a segmentation upon completion, as well as toggle its visibility.

AI Box Segmentation. The box segmentation tool expects a user to draw a tight bounding box around an object of interest, where the algorithm (based on [24, 42]) outputs a single closed polygon that encloses the salient object within the bounding box. While it is possible to generate multiple polygons that better represent the object, deal with holes etc., a more predictable tool is easier to interact with in our experience.

AI Trace Segmentation. This tools expects a user to scribble a quick and coarse boundary around the object, where the algorithm outputs a single closed polygon that captures fine details and "snaps" to the boundaries of the object of interest.

AI Correction. When a user corrects the position of a single vertex in a polygon, we algorithmically predict offsets for vertices in a local neighbourhood of the corrected vertex. Once a vertex is corrected by a human, it is marked such that subsequent AI interactions do not displace it.

AI Refinement. The refinement tool allows "snapping" any polygon to better fit it to object boundaries.

Path Correction. For correcting large errors, users can draw a new polyline or scribble connecting any two vertices of an existing polygon. The newly drawn polyline or scribble is "snapped" to boundaries using the same algorithm as in AI Refinement.

Paint/Erase. This tool allows users to draw freeform segmentations using a circular brush with adjustable radius. This freeform segmentation can be used to draw a new polygon, add to an existing polygon or subtract from an existing polygon.

Segmentation Booleans. Segmentation booleans have multiple uses. Any polygon can be subtracted from or merged with any other polygon. This allows annotators to draw holes, reuse precise boundaries drawn for a different object adjacent to the current object of interest, and deal with occlusions faster.

D.2 Project Management

Managing annotations for a large dataset requires scalable methods to assign, monitor and review annotation tasks. For this purpose, we extensively use the python API accompanying TORAS. All task management was done using python scripts and progress was monitored through a slack bot in a shared workspace. Comments obtained from stage3 were sent back to annotators using the API, which are shown clearly in the task UI. Annotators were able to see whether a task has a comment using a special indicator in their list of annotation tasks.

Table 7: **Inter-annotator agreement.** We report mean intersection-over-union (IoU) of segmentations from five different annotators on a subset of 186 images with 940 objects. Our results show a high average inter-annotator agreement of 90.3 IoU, consistent with OpenImages [5]

	Ann. 1	Ann. 2	Ann. 3	Ann. 4	Ann. 5
Ann. 1		88.81	89.51	90.39	90.76
Ann. 2	88.81		89.56	89.28	90.45
Ann. 3	89.51	89.56		90.98	92.30
Ann. 4	90.39	89.28	90.98		91.49
Ann. 5	90.76	90.45	92.30	91.49	

D.3 Annotator Agreement

To verify annotation quality, five of our annotators annotated the same set of 186 images independently. These 186 images had 1422 total objects listed for annotation, obtained from our over-complete entity identification stage. Out of the 1422 objects, 940 objects were annotated by all five annotators. We compute segmentation agreement on these 940 objects in Table 7 and report 90.3 average pairwise mean IoU between all annotators (i.e. averaged from mean IoU of 10 combinations of two annotators from our group of five). This is in line with the 90 IoU human agreement per instance reported for OpenImages [5] and much higher than the ≈80 IoU agreement reported for MS-COCO polygons in the OpenImages paper [5].

Out of the 1422 objects, 940 were annotated by all five annotators. Out of the 482 not annotated by all, 295 were left empty by all annotators i.e. they consistently agreed that the object does not appear in the image. Thus, 13% (187 out of 1422) of the total objects in this subset were inconsistently annotated. Of the 187 remaining objects, 97 were annotated by four (out of five) annotators, 22 by three, 30 by two and 38 by one annotator. We note that our subsequent correction annotation aims to fix these errors of missed objects (among other errors).

D.4 Annotation Efficiency

We measured annotation speed-up using our AI tools over manual annotation, by annotating a subset of 50 images from different videos, containing 253 objects. One of our trained annotators annotated these 50 images with all of our tools (including AI tools) on one day, and manually on another day. Only using manual tools resulted in an average time of 50.5 seconds / object, while obtaining 89.92 IoU agreement with our released ground truth. With access to AI tools, annotation time went down to an average of 37.5 seconds / object, i.e. at approximately three quarters of the time, while obtaining 90.11 IoU agreement. The IoUs obtained are within one standard deviation of each other. Thus, on this subset, we verify that there is no accuracy degradation with using AI tools for segmentation, while saving a quarter of the time over manual segmentation. We also find that our annotators got faster through the project, showing up to 2x speed improvement in number of images annotated per hour.

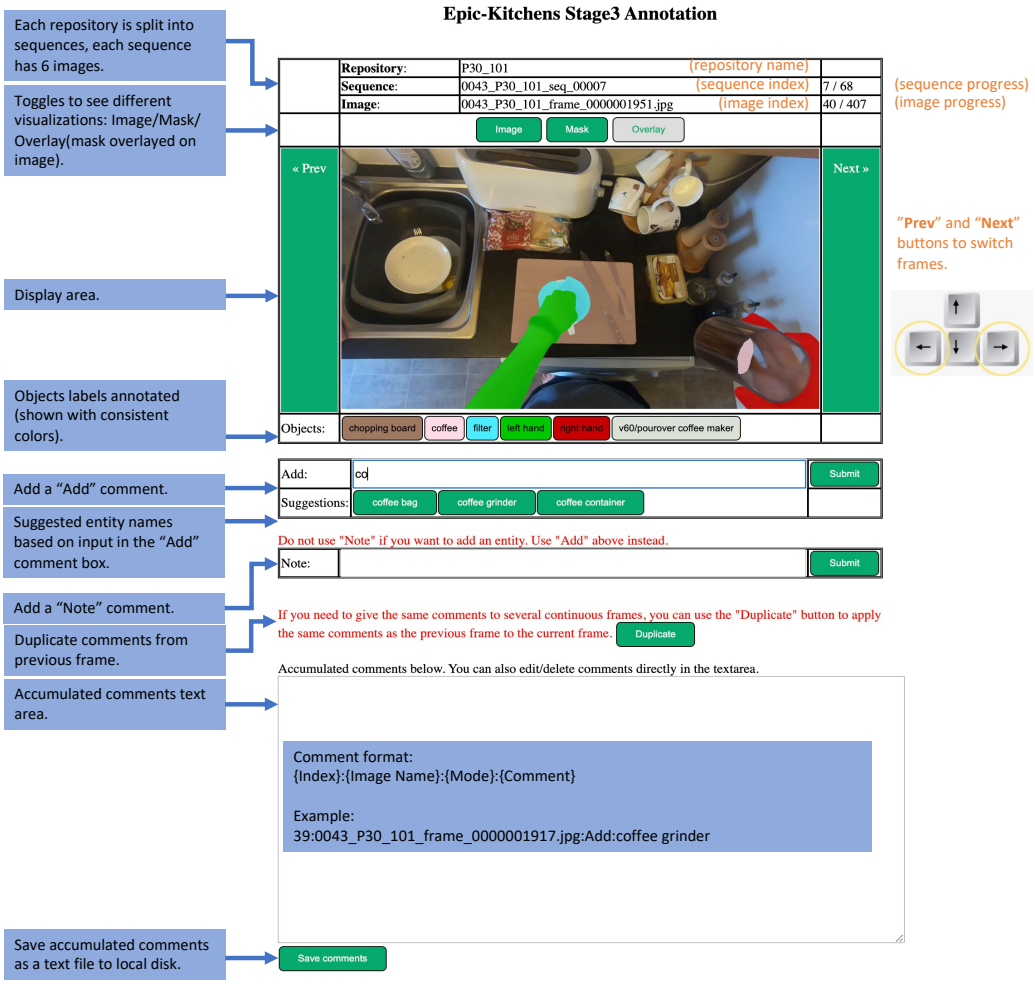


Figure 17: **Correction stage user interface for writing comments for each image**. For each annotation repository (i.e., sequence of frames given to the annotator), we generate a HTML page to show its annotation from the pixel labels.

E Appendix – Correction (Main §2.2) / Correction

The goal of our correction stage covering all previous annotations is to make sure the masks are accurate within the current frame and sequence. We plot out all annotations on images and then manually scroll through the frames, in order to confirm/correct consistency. We describe our interface (§E.1), how the corrections are performed on TORAS (§E.2), and statistics about the corrections (§E.3).

E.1 Correction Interface for Collecting Comments

The goal of our correction interface is to spot annotation errors in pixel-level segmentations. First, we ask commenters to give comments for each frame using the interface in Fig. 17. The interface contains three blocks, display block, commenting block and comments block.

Display. On the top, the display block shows information about the current repository, sequence and image as well as indicating the sequence progress and image progress in text to let the commenters know their progress. In the middle, the current image/mask/overlay(masks plotted on the image) as displayed in the center, with toggle button to choose which mode (Image/Mask/Overlay) to show. With Overlay set as the default toggle, the Image/Mask display is very helpful when the overlaid annotation makes it hard to check some details. The prev/next button is to switch between images

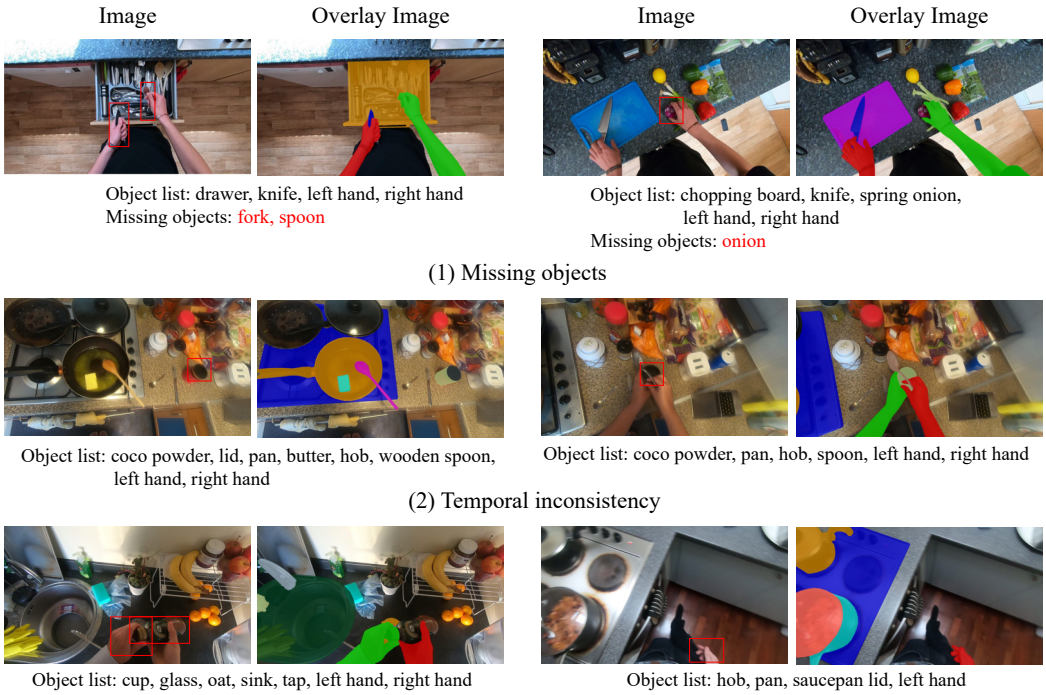


Figure 18: **Typical errors that are fixed in the correction stage**. We show three kinds: missing objects, temporal inconsistency, and incorrect annotation.

which can also done by using the left/right key on the keyboard, enabling swift switching. On the bottom, the object list shows the entity names highlighted with corresponding colors as in the overlaid display.

Add/Note/Duplicate. We invite the corrector to provide any of three types of corrections. First, the "Add" mode allows choosing from the suggested entity names by clicking. There is a matching algorithm behind this suggestion technique. Candidate entities come from other entities in the same video. New entity names can also entered. The second "Note" mode is used mainly for submitting comments about renaming (e.g., "rename paper to carton", "rename mixture to pancake") and correcting existing annotations (e.g., "missing part of the sink segmentation", "table is incorrectly segmented (missing corner)"). The last one is "Duplicate" which copies comments from the previous frame to the current frame, as some errors persist in more than a single image, thus accelerating the correction stage.

The comments are aggregated in the text box listed in reverse order. Once the repository is done, the comments can be saved as a text file locally.

E.2 Correction on TORAS

We upload comments to TORAS for annotators to do correction for all the repositories. The comments are shown for each frame and the annotators are asked to correct and resubmit images.

For the Train and Val sets, we typically used a single round of corrections, except where a video was marked for a double-round of corrections. For the Test set, all videos passed through two rounds of corrections.

E.3 Statistics of Collected Comments

Fig. 18 shows three typical kinds of errors detected in this correction stage, namely missing objects, temporal inconsistencies and incorrect annotations. First, missing objects are the most common error, which occur when some active objects do not exist in the object list for annotation. Two examples of missing objects are presented in Fig. 18 (1). In the left example, "fork" and "spoon" are missing

while “onion” is missing in the right example. Second, temporal inconsistencies refer to annotations in consecutive frames or sequences that are not consistent across sequences. In Fig. 18 (2), the “lid” of “coco powder” is annotated in the left frame, but missing from the right. If the two frames are examined independently and temporal consistency is not taken into account, both annotations can be regarded as correct. Third, incorrect annotations, such as Fig. 18 (3) left, show the “lid” of the “syrup bottle” segmented as part of the “left hand”, with right showing the “right hand” annotated as “left hand”.

Regarding the statistics on the number of images that were passed back for corrections, we report that 12,442 images out of our 50.7K images were passed back with at least one comment. The proportion of images that needed correction is thus 24.5%. The proportion reported above is of all the images. On average, images that required correction had 1.8 comments from the manual verification stage. Of these, 80% of the comments requested an addition (or renaming) of an active object, while 20% of the comments were related to refining the segmentation boundary.

F Appendix - VISOR Object Relations and Entities (Main §2.3)

Our object relations and exhaustive annotations are obtained from a crowdsourcing company (Hive aka thehive.ai). This section describes the quality controls done by this platform (§F.1) as well as the annotation instructions for Hand Object Segmentation (§F.2) and Exhaustive Annotation (§F.3). We hired Hive to provide the annotation, and they in turn hired annotators, so translating directly into an hourly rate is difficult. However, we paid $20 per thousand instances labelled for both tasks.

F.1 Quality Control

Hive implements standard quality control during the annotation process consisting of qualifiers (tests before annotation), gold standard checks (tests during annotation), and consensus labelling (aggregation of multiple judgements).

Qualifiers. Before workers start a task, they are asked to complete a qualifier. This qualifier explains the annotation task and administers a test that workers must pass before they start the process. For all annotations, we required workers to achieve at least an 80% on the qualifier test in order to start annotating.

Gold Standard Checks. Throughout the annotation process, workers are shown questions with known answers (usually clearcut cases). Workers that do not perform accurately on these known cases are dismissed. This helps catch guess-work efforts. For all annotations, we required workers to maintain a 90% accuracy rate on known samples to continue annotating.

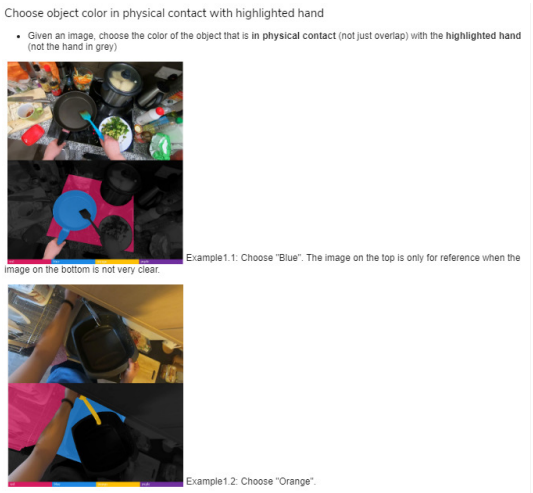
Annotator Consensus. Each instance was annotated by multiple annotators. A data instance was only considered labelled if six out of (up to) nine workers agreed on its label. Data in which this consensus could not be obtained were marked as inconclusive.

F.2 Annotation Instructions for Hand-Object Segment Relations

We show the annotation instructions for the Hand-Object Relations task in Fig. 19. To give a consistent interface regardless of which object the hand is contacting, we colour potentially in-contact objects. We define potentially in-contact objects as any segmented object that shares a border with the hand. A small number (295) of frames have more than four potentially-in-contact objects. We annotate these manually ourselves.

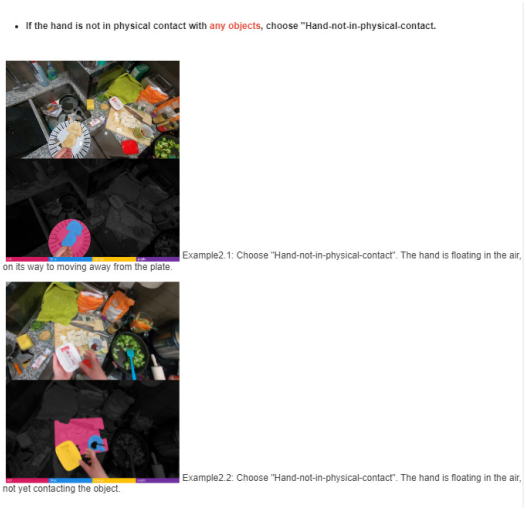
F.3 Annotation Instructions for Exhaustive Annotation

We show the annotation instructions for the exhaustive annotations in Fig. 20. Entity name is provided in the top right corner of each image, and triplet boundaries with three different colours are used to highlight each entity with a visual cue to avoid confusion. As shown in the figure, the annotators should give a binary decision on whether all pixels related to the class listed on the top-right have been segmented. Consistent annotations are used in the exhaustive flag.

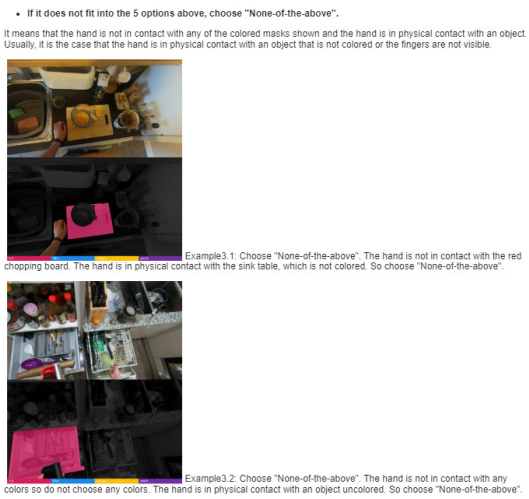


Page 1 – General Case

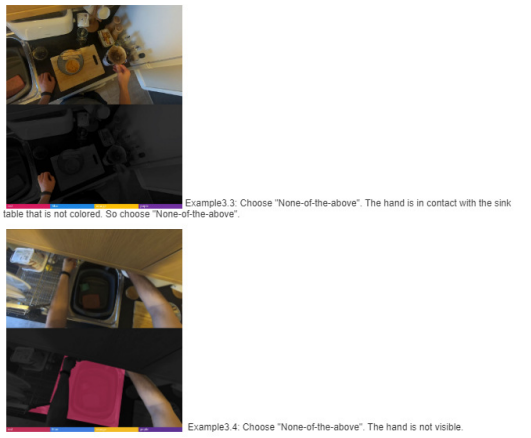
Page 2 – General Case



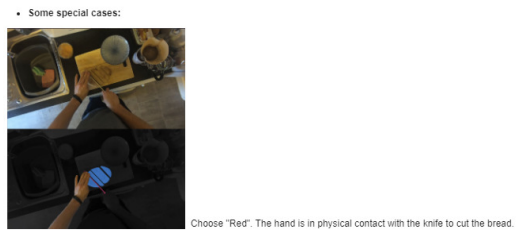
Page 3 – Not in Contact



Page 4 – None of the Above



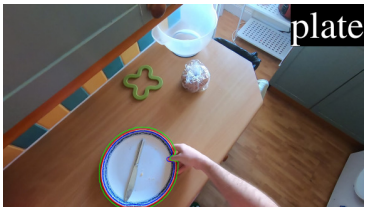
Page 5 – None of the above



Page 6 – One Edge Case

Figure 19: **Annotation Instructions for the HOS Task in Hive**. These cover several cases: a general case, not in contact, non-of-the-above, and a common edge case.

Given an image, answer the question "**Are ALL of the top-right item(s) contained in colorful boundary?**"



Choose 'Y'.

Choose 'N'. There is another plate with no colorful boundary in the image.

Choose "Y".

Choose 'N'. The bananas on the shelf in the right corner of the image are not highlighted.

Figure 20: **Annotation Instructions for exhaustive annotations in Hive**. We outline the object in question with colorful boundaries.

Table 8: **Detailed statistics of the automatic interpolations.** The numbers include the 2 annotated frames of each interpolation

	Before filtering	After filtering
# interpolations	35.2K	32.2K
# interpolated objects	172.0K	119.4K
# images	3.2M	2.8M
# masks	14.7M	9.9M

G Appendix - Dense Annotations (Main §2.4)

Interpolation details. We use STM model, pretrained on MS-COCO, and fine-tune it on VISOR annotations. We use 5 memory frames during inference. For evaluation, we use one memory frame to reconstruct the annotated frames for faster inference. Our code uses three Tesla V100 GPUs to generate the interpolation, two of them are used for the forward and backward STM passes, and one for combining the logits and evaluating the interpolations.

Interpolation statistics. Table 8 shows the full statistics of the interpolation before and after filtering, we filter out interpolated objects with $\mathcal{J}\&\mathcal{F} < 85\%$ as shown in Fig. 22, so we keep 69.4% of the object interpolations, having 9.9M masks. Fig. 21 shows the distribution of the of the interpolated segmentations per class, the distribution is similar to the sparse one in Fig. 4. Hands masks form 40% of all filtered interpolations.

Visualisations. We show sample automatic dense annotations at 20%, 40%, 60% and 80% of the interpolated sequence length in Fig. 23 which includes objects with different sizes and segmentation challenges. The figure shows how accurate the interpolations could be as we use information from the two annotated frames. Fig. 24 and Fig. 25 show more detailed visualizations of the interpolations to showcase the temporal information they provide. We plot a frame every 5 frames for two long interpolation sequences. In Fig. 24, we show butter being spread on toast where the knife, jar, butter, chopping board, toast and both hands are accurately segmented. In Fig. 25, we show very fast motion as a knife is sharpened. Some masks are inaccurate like frame because of the severe motion blur. Additional examples in the form of videos can be found on the project webpage.

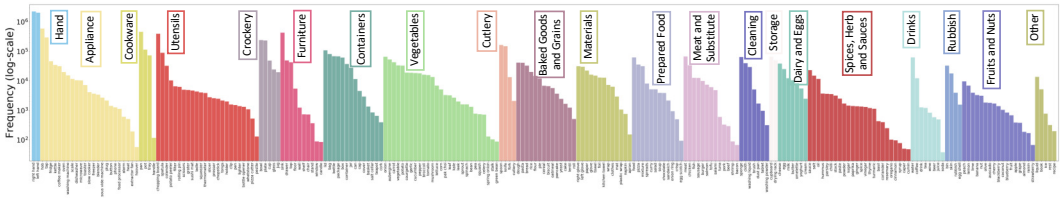


Figure 21: **Frequency of interpolated entity classes** (Log y-axis) The histogram is long tailed. Best view with zoom

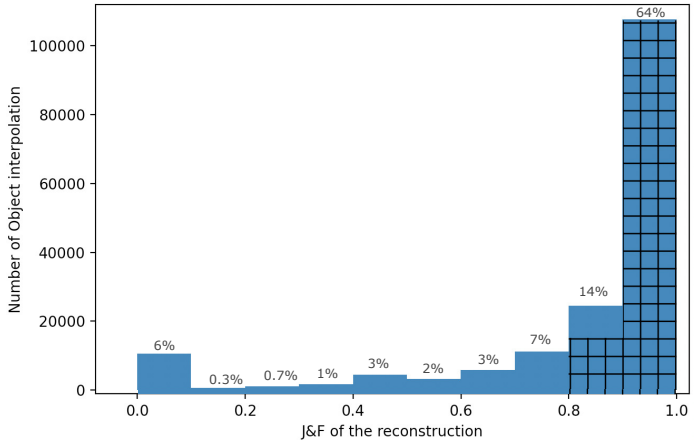


Figure 22: **Histogram of interpolations' scores.** The hatched area is the selected area after filtering any $\mathcal{J}\&\mathcal{F}$ scores that are less 85%.

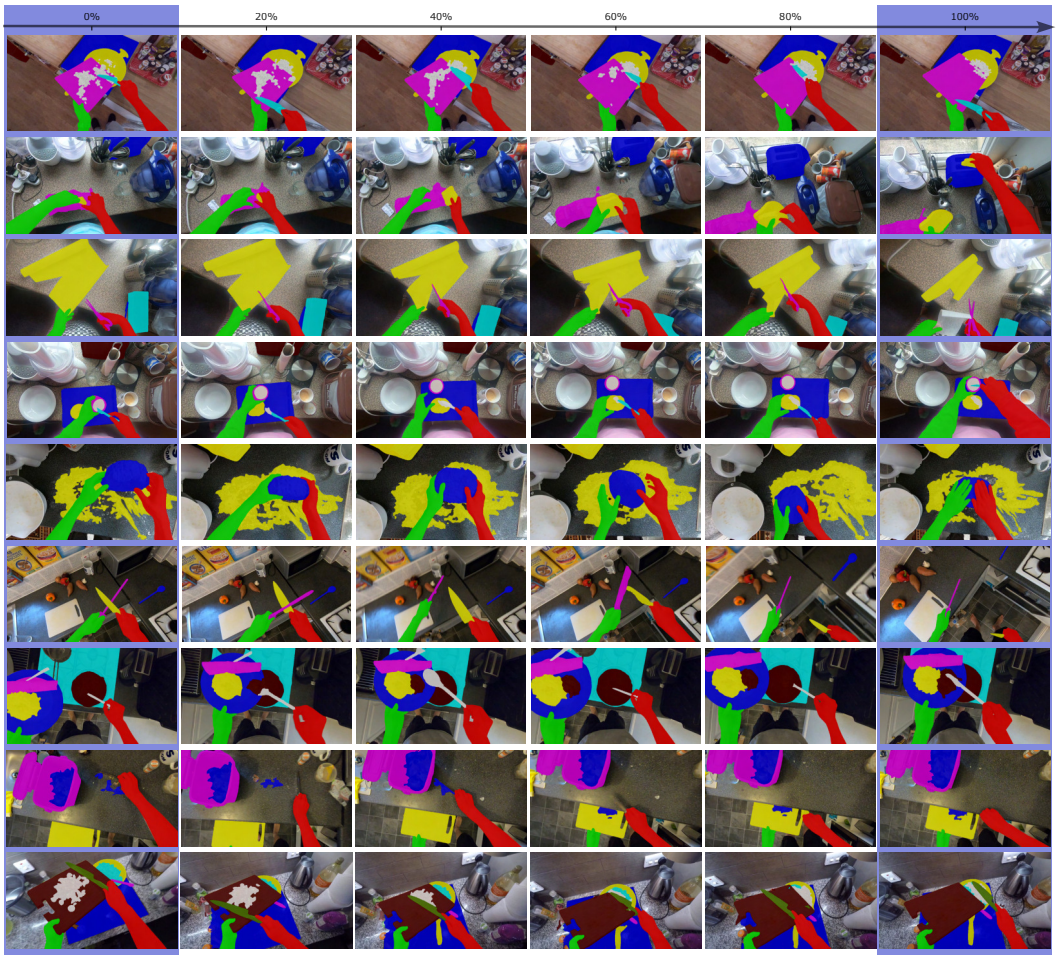


Figure 23: **Sample segmentations from the dense automatic annotations.** First and last columns show the ground truth manually annotated frames and the rest are samples from the automatic interpolations at different ratio of the interpolation length.

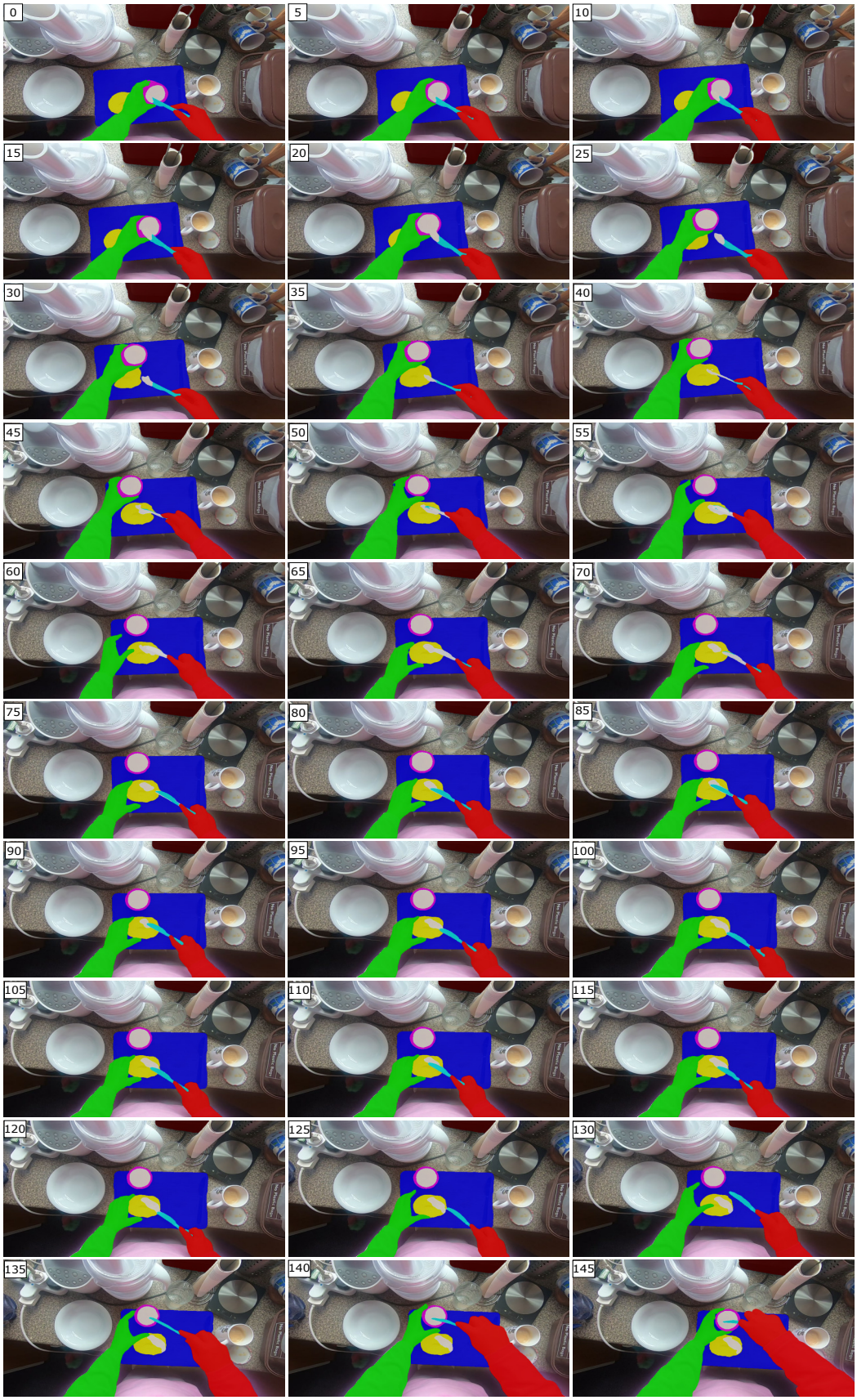


Figure 24: **Example dense segmentations from the dense automatic annotations sampled every 5 frames.** The frame number included in the top-right corner of each frame

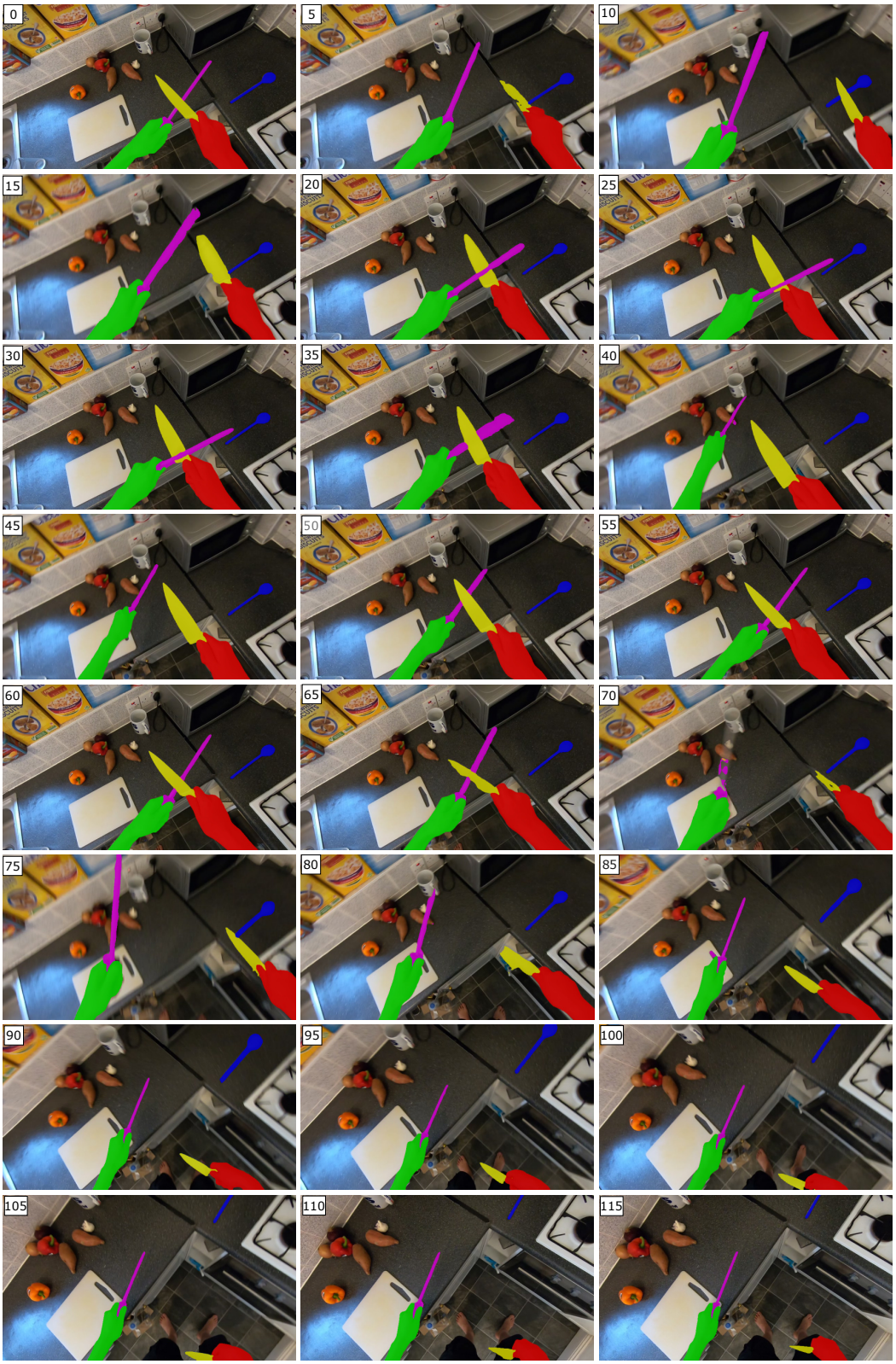


Figure 25: **Second example dense from the dense automatic annotations sampled every 5 frames.** The frame number included in the top right corner of each frame

Table 9: **Statistics of VISOR for semi-supervised VOS.**

	Kitchens/unseen	Seq	Masks	Obj/Seq μ/σ, min, max	Img/Seq μ/σ, min, max	Cls/unseen
Train	33	5,309	174,108	5.2 / 2.1, 1, 13	6.2 / 3.4, 3, 171	242
Val	24 / 5	1,244	34,160	5.3 / 2.3, 1, 13	6.0 / 3.5, 2, 93	165 / 9
Test	13 / 4	1,283	46,998	5.4 / 1.8, 1, 10	7.7 / 9.3, 2, 186	151 / 6

H Appendix - VOS Benchmark Details (Main §4.1)

This section describes the semi-supervised VOS Benchmark, including data preparation (§H.1), metrics and evaluation (§H.2), baseline training details (§H.3), and additional results (§H.4).

H.1 Data Preparation

We adapt splits to be suitable for this benckmark. For each subsequence in Val/Test: we keep objects that appear in the first frame and ignore others from evaluation. This is because the benchmark only tracks segmentations that are present in the first frame. For the train set, we have not ignored any masks, we have just ignored any subsequence with less than 3 frames since at least 3 frames are required to train the network. Table 9 shows the statistics of the adopted version for semi-supervised VOS. We highlight unseen kitchens in Val/Train as well as unseen (or zero-shot) classes.

H.2 Metrics and Evaluation

We report our results in this benchmark using two measures as proposed in [32]: **(1) Jaccard Index ($\mathcal{J}$)** Given the ground-truth G and predicted mask P, the Jaccard Index is defined as $\mathcal{J} = (|P \cap G|)/|P \cup G|)$. This metric is not particularly sensitive to boundary accuracy, but is a common metric for evaluating segmentation. **(2) Boundary F-Measure ($\mathcal{F}$)** Given the ground-truth and predicted contours, the boundary F-measure $\mathcal{F}$ is defined as the F-score of the precision (P) and recall (R), or $(2PR)/(PR)$. This metric is more sensitive to boundary accuracy.

The final score is the mean of the two metrics for all subsequences, and the score for each subsequence is the mean of its constituent objects. For full details refer to *§3* in [32].

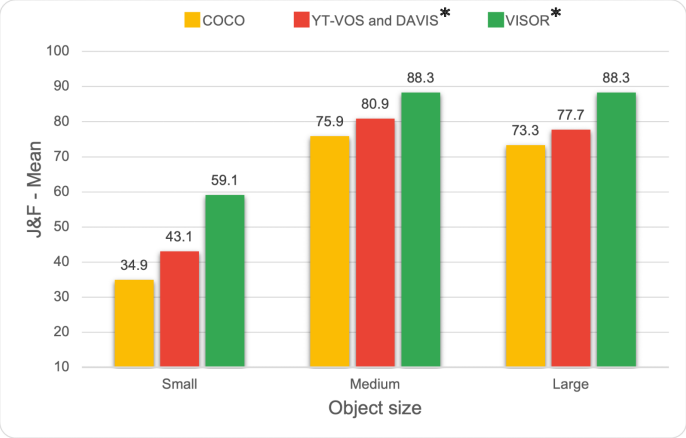


Figure 26: **Size-based performance of STM trained on different datasets**. * means the model is pretrained on COCO. $\mathcal{J}\&\mathcal{F}$ is calculated on VISOR Val set and averaged on all object masks of each object size category (small, medium and large).Small objects are harder to be segmented

H.3 Baseline Training Details

Training Details. To train the baseline, we sampled 3 random images resized to 480p (854x480) from a subsequence with dynamic skipping between them. As we have sparse annotations with a dynamic number of intermediate frames, we initially sample without skipping and then use a maximum dynamic skipping of 1 half-way through the training process in a curriculum learning fashion.

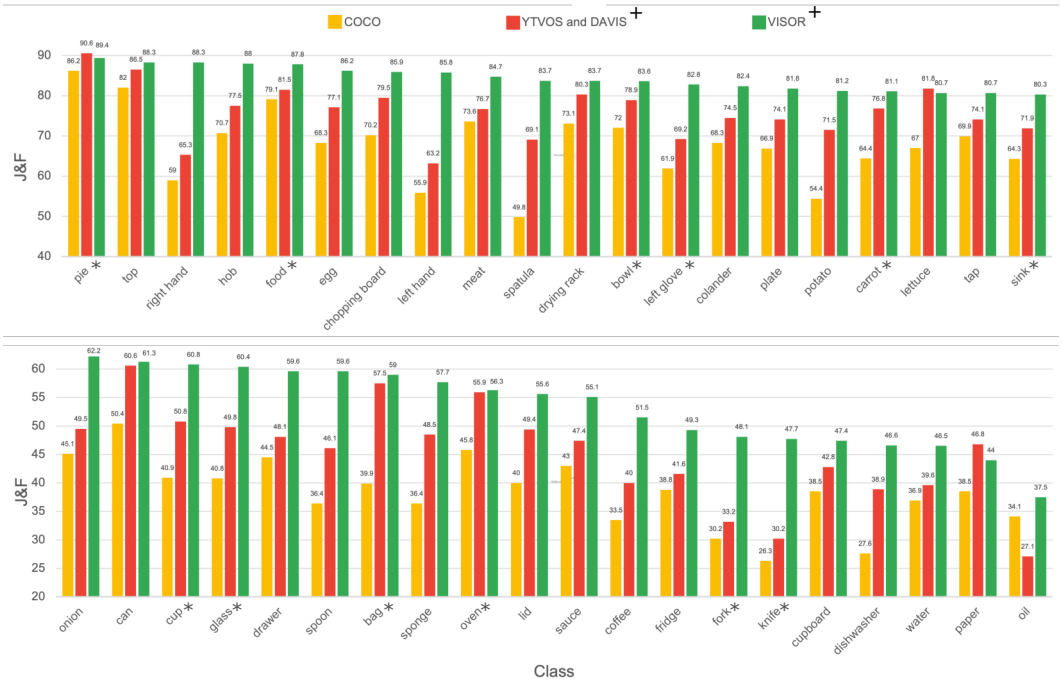


Figure 27: **Class-based performance of STM trained on different datasets**. + means the model is pretrined on COCO, * means that the class is part of COCO categories.

We use Resenet-50 as backbone. We train using a fixed learning rate of 10^{-5}, batch of 32 and 400,000 iterations to fine-tune the COCO pretrained model. We also use cross-entropy loss and the Adam optimizer. We use single Tesla V100 GPU to train and it took 4 days to train on our dataset.

During inference, since we have multiple objects per sequence as mentioned in Table 9 we segment each of them then we use their logits to classify the class for each pixel. Also, since the average number of the sparse frames per sequence is low (6.0 and 7.7) frames for validation and test set respectively as mentioned in Table 9, there is no need to keep frames every N frames. We report results using 2 memory frames sampled uniformly throughout the sequence. We have just evaluated the model using the sparse annotations, however, denser frames could be extracted from EPIC-KITCHENS-100 videos (i.e. 50fps) to use during inference, this may help to get better results, but it is time and memory costly.

H.4 Additional Results

In addition of the results reported in the main paper, we provide more detailed results, breaking down results by size and class.

Size-based performance. In Fig. 26 we calculated the object size of the ground-truth masks of val, then we splitted them equally into small, medium and large objects. The figure shows that all models suffer with small objects, but the model fine-tuned on VISOR is 16% better than the one pretrained on COCO and fine-tuned on both YTVOS and DAVIS. This gap is reduced to 7.4% and 10.6% for medium and large respectively.

Class-based performance. Fig. 27 shows the best and worst performing EPIC-KITCHENS-100 classes for 3 different models. The gap between the model fine-tuned on VISOR and others is not too large for the classes that are part of COCO dataset such as pie, food whereas the left/right hands and spatula have largest margins. In the worst classes, knife and fork have poor scores since they usually change their appearance during the subsequnece (based on the view point and object orientation) and as tools they usually are occluded most of the time as part of their functionality.

Table 10: **Hand Object Relation Annotation Stats.** Most of the annotations are marked as being in contact with one object.

	Train+Val	Test
In Contact	52,685 (81.2%)	14,233 (82.4%)
Not-in-contact	4,144 (6.4%)	1,341 (7.8%)
None-of-the-above	3,079 (4.7%)	592 (3.4%)
Inconclusive	4,943 (7.6%)	1,104 (6.4%)
Failed upload	1	1
> 4 candidate objects	272	23

Table 11: **Gloves-on-which-hands annotation distribution.**

	Glove
on left hand	497
on right hand	561
on both hands	13
not on hand	304
inconclusive	34
total	1409

Table 12: **Glove object relation annotation distribution.**

	Glove-on-Hand
glove-in-contact	930
glove-not-in-contact	31
none-of-the-above	9
inconclusive	101
total	1071

I Appendix - HOS Benchmark Details (Main §4.2)

We now describe the HOS Benchmark, including data preparation (§I.1), metrics and evaluation (§I.2), baseline training details (§I.3), and additional results (§I.4).

I.1 Data Preparation

Hand Object Relation Annotations from Hive. The annotated labels for hand object relations from Hive are shown in Table 10. "Failed upload" means samples that failed uploading to Hive, which we manually annotated. "More than 4 candidate" cases are samples that have more than 4 candidate masks potentially in contact with the hand. Since we only showed the annotators 4 colored options, we manually checked and annotated these ourselves.

Hand in Gloves. As noted before, hands are frequently in gloves during some kitchen activities like cleaning or using the oven. When the glove is worn on a hand, we consider the glove as part of the hand, which means the current hand mask is now any visible hand parts plus the glove mask. The object that the glove is in contact with is thus considered as an object in contact with the hand. Gloves that are not worn on hands are considered as normal objects/masks.

In the data set, there are 941 (674/247/20 in train/val/test set) images with 1,409 (957/432/20) glove entities, out of which 1,105 (698/396/11) gloves are on hands. For hands in gloves, we follow the same way to annotate the in-contact objects as we did with hands. Table 11 shows distribution of glove annotations and Table 12 shows the distribution of the annotated glove-object relations. Some examples of glove object relations are shown in Fig 28.

Training and Evaluation Data. We prepare our data annotations in COCO format for each task separately following the same train/val/test split of VISOR.

For *Hand-Contact-Relations* task, we prepare two-class annotations, one class for "hand" (both left and right hand), the other one for "object" which are the contacted objects annotated in the VISOR Object Relation annotations. Additionally we add: hand side, in-contact and offset, in the original COCO annotation. Hand side is binary where 0 indicates left hand and 1 indicates right hand. Contact is also binary, where 0 indicates the hand is not in contact and 1 indicates the hand is in contact. The offset from the hand bounding box centre to the in-contact object bounding box centre is factored into a unit vector $v \in \mathbb{R}^2$ and a magnitude $m \in \mathbb{R}$, as in [A]

For *Hand-And-Active-Objects* task, we prepare two-class annotations, one class for "hand" (both left and right hand), the other one for "object" which are all other objects.

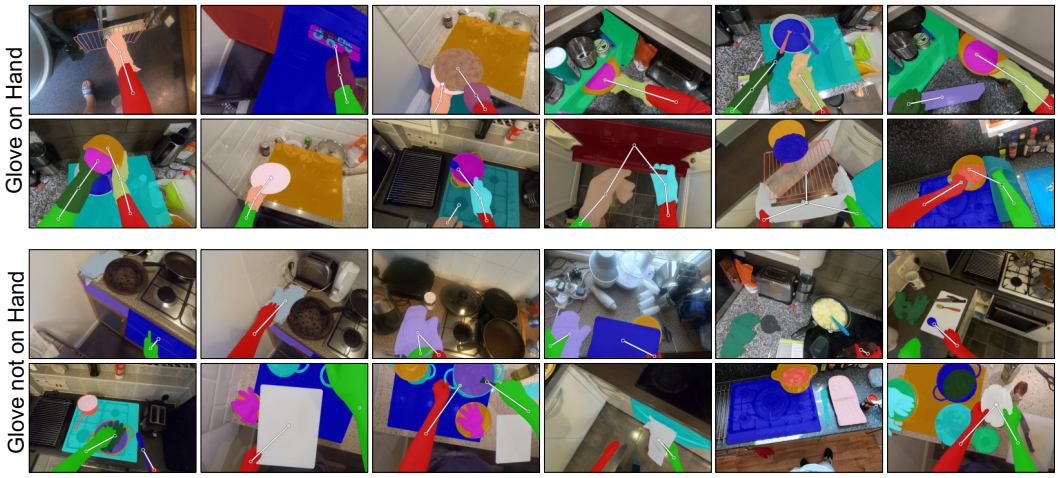


Figure 28: **Glove object relation annotation**. Glove-on-hand examples are shown on the top two rows and Glove-not-on-hand examples are shown on the bottom two rows.

I.2 Metrics and Evaluation

We evaluate via instance segmentation tasks using the COCO Mask AP [23]. We evaluate per-class to better show the performance on each class. We only keep images with conclusive annotations in Val/Tes. In other words, if there is a hand in the image that has inconclusive or none-of-the-above annotation for contact state we do not use it in evaluation.

For *Hand-Contact-Relations* task, we prepare three schemes for hand classes: all hands as one class; hands split by side (left/right); and hands split by contact state. These 3 individual evaluations better show the performance on purely hand mask prediction, hand mask + hand side prediction and hand mask + contact prediction. The offset between hand and object is evaluated implicitly by associating each hand that is predicted as in-contact with an object if exists. Here the object evaluation is on the object prediction after this post-processing.

For *Hand-And-Active-Objects* task, we do normal per-class instance segmentation evaluation.

I.3 Baselines and Training Details

For both tasks, we use PointRend [21] instance segmentation model implemented in Detectron2 [B] with R50-FPN backbone and standard 1× learning rate scheduled configuration by default. It is trained for 90,000 iterations with batch size of 24 and base learning rate of 0.02.

Specifically for *Hand-Contact-Relations* task, we add 3 additional linear layers after the ROI-Pooled feature to predict hand side (feature size, 2), contact state (feature size, 2) and offset (feature size, 3). During training, we use Cross-Entropy loss for hand side and contact state and MSE loss for offset. We skip and do not supervise hand contact and offset on inconclusive and none-of-the-above hand contact annotations.

I.4 Additional Results

We showcase additional HOS qualitative results on validation set in Fig 29.

Figure 29: **Qualitative results of HOS on Val set**. Hand segments are often very accurate and many objects are segmented correctly, although this is still a challenging task.

J Appendix - WDTCF Benchmark Details (Main §4.3)

In this section, we introduce the data annotation (§J.1), evaluation (§J.2) and baseline details (§J.3) and additional results (§J.4) for the WDTCF benchmark.

J.1 Data Preparation and Annotation

First, we select query object candidates that are meaningful to ask the question "Where did this come from?", for instance, static objects (e.g., 'fridge', 'oven' and 'sink'), hands, mixture and food are excluded. Then, we extract the query and evidence frame candidates based on rigorous rules. Concretely, given an untrimmed video, the frames for each query object are linked throughout, (e.g., 'bowl' and 'milk'). We consider the last three frames that feature the query object as potential query frame candidates, and the first three frames with co-occurrence with the object and any of our 15 containers as evidence frame candidates. The number of query and evidence frames are empirically set to be three.

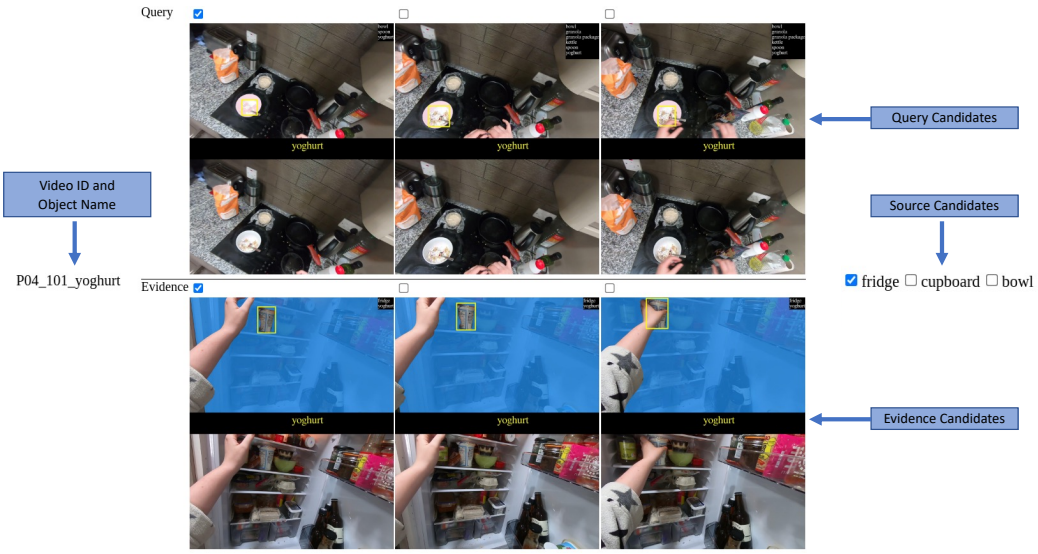


Figure 30: **Annotation interface for where did this come from**. This interface is used to identify the source object in the video.

Fig. 30 shows the interface for annotation. The key components contain the video ID, the query object name, query candidates, evidence candidates and source candidates. In this example, the query object 'yogurt' can be clearly seen emerging from 'fridge'. Note that although it could contain multiple evidence frames for each query object, only one is annotated for the taster challenge.

We annotate 222 examples for this test set. Fig. 31 shows the distribution of duration between query and evidence frames. While many gaps are small (within 2 minutes), the duration varies greatly, with 19.4% longer than 10 minutes.

J.2 Evaluation Details

Given the query segment, each method produces: (1) a class id indicating the source object; (2) an evidence frame where the query object emerges from the source object; as well as (3) segments of the source and query objects in the evidence frame.

This challenge is defined on sparse frames. While methods are free to look at the dense frames, they are asked only to produce results and evaluated *only* on the sparsely annotated frames. This is important to make sure that methods are not being asked to identify the *precise* moment that an object emerges out of a high frame rate video. Instead, WDTCF asks the method to identify which of a number of distinct keyframe best shows the object emerging.

Each method is evaluated on three metrics. First, *Source* evaluates the accuracy of the class prediction (i.e., whether the predicted class is the same as the ground-truth class). Second, *Query IoU* evaluates

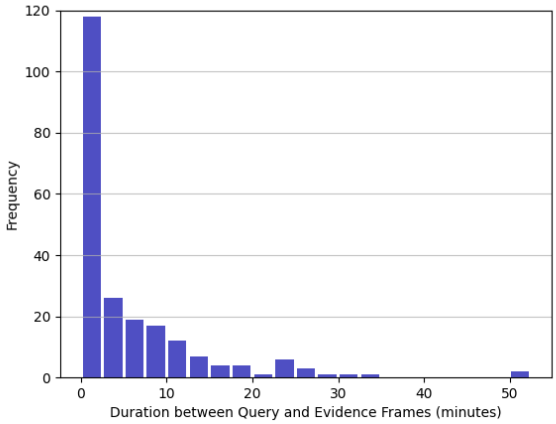


Figure 31: **Distribution of duration between query and evidence frames.** Some are close, but many are far away.

the intersection-over-union/Jaccard Index of the query object. This is zero if the evidence frame is not localized correctly. Finally *Source IoU* evaluates the IoU of the source object. This is also zero if the evidence frame is not localized correctly.

J.3 Baseline Details

We next explain how we use PointRend model to produce baseline results with oracle knowledge which is also trained with R50-FPN backbone and default learning schedule in Detectron2 [B] for 90,000 iterations using batch size 24 and base learning rate 0.02. Note that the model is trained using the Train and Val sets of VISOR, and the WDTCF examples are obtained from the Train and Val sets as well. We predict the query object class from the best overlap with masks based on the trained model. Note that WDTCF is defined only on the sparse frames with ground truth masks, which we use for IoU evaluation. Specifically, if the query name is predicted correctly, then the model is used to further detect the co-occurrence of the object and one of the potential 15 sources starting from the first frame of the video as evidence frame candidates. We consider first-3 candidate frames and pick the one with the highest confidence score as evidence frame prediction.

In baselines where the evidence frame is given, the problem is simplified to predicting the source in the evidence frame, regardless of the query object. Consequently, the prediction of query and source masks of the evidence frame are directly used to compute the IoU with ground truth masks.

Query Object Prediction. Fig. 32 shows two examples of PointRend prediction results on query frame. In the left example, the query object is detected and segmented perfectly. Therefore, the query object 'bottle' can be predicted correctly by comparison with the GT masks of 'bottle'. However, the 'yogurt' in the bowl is undetected due to occlusion of cereal in the right example. For all the query masks, the overall prediction accuracy is 90.5%, which shows that the query object segmentation is non-trivial. For instance, the tiny objects (e.g., garlic) and liquid (e.g., coffee and oil) are hardly to be detected.

Evidence Frame Segmentation. Fig. 33 shows two examples of PointRend segmentation results in the evidence frame. The only predicted source in the left example is 'fridge', thus it is trivial to predict the source with given evidence frame. In contrast, there are two source candidates are detected in the right example, i.e., 'bottle' and 'cupboard'. Without prior knowledge, it is difficult for the model to decide which one is a better choice.

J.4 Additional Results

We showcase additional WDTCF qualitative results in Fig 34.

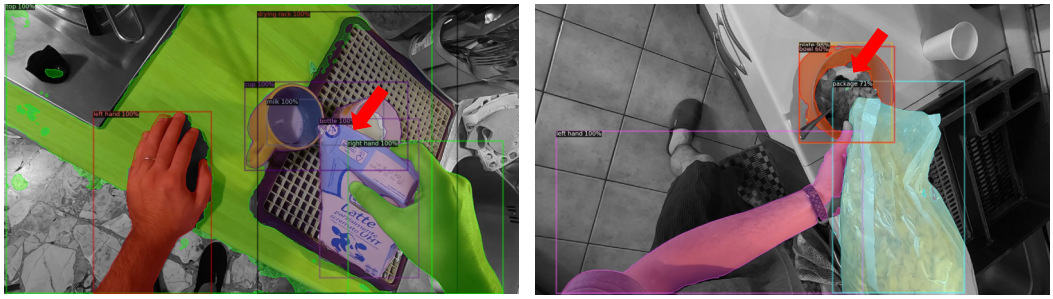


Figure 32: **Two examples of PointRend prediction results on query frame**. We show one success (left with query object 'bottle') and one failure (right with query object 'yogurt').

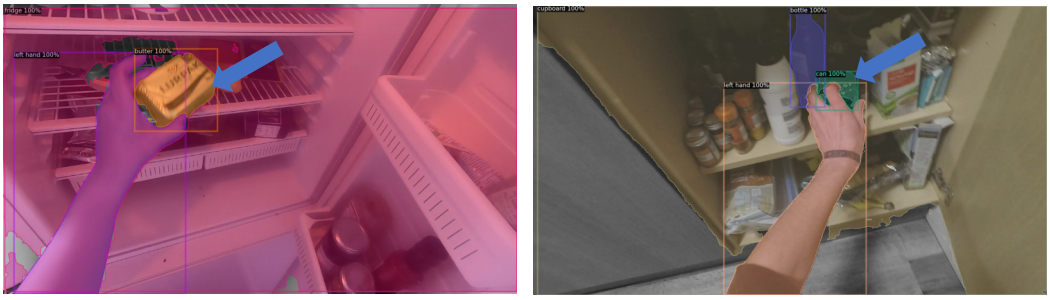


Figure 33: **Two examples of PointRend prediction results on evidence frame.** The blue arrow indicates the query object.

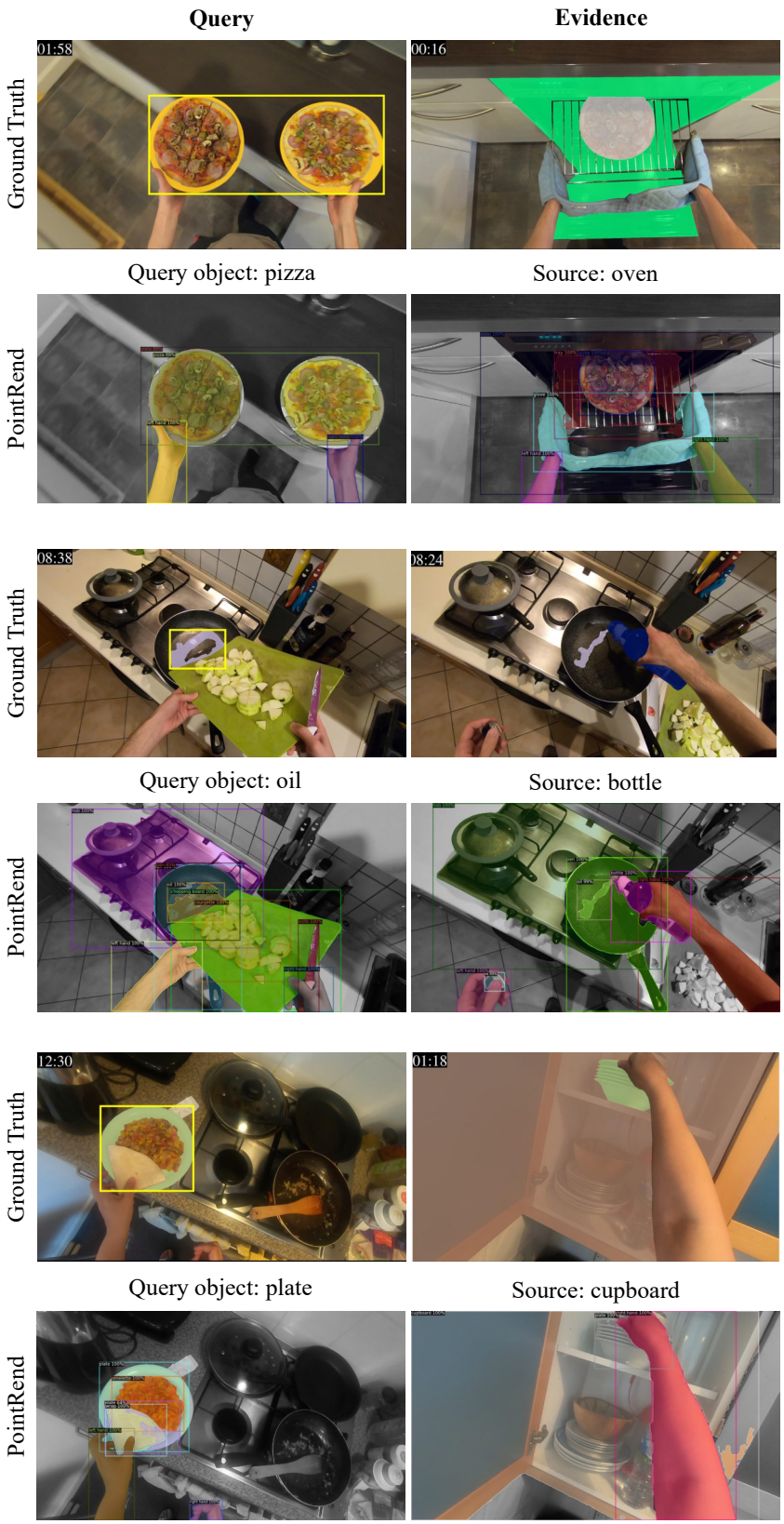


Figure 34: **Visualization of ground truth along with PointRend prediction results.**

K Appendix - EPIC-KITCHENS VISOR - Datasheet for Dataset

This paper introduces a new set of annotations, VISOR, for the EPIC-KITCHENS-100 dataset. Given that the VISOR annotations depend on the EPIC-KITCHENS-100 dataset, we have also answered some of the questions about ethics and consent for EPIC-KITCHENS-100. We have marked these answers in blue. For instance:

Q. Datasheet Question

A. This answer applies to VISOR

(EK) A. This answer applies to EPIC-KITCHENS.

Motivation

Q. For what purpose was the dataset created? *Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*

A. The VISOR dataset introduces a large dataset of segmentations that are consistent segmented throughout long videos, as well as relations between these segments in both time and space. The goal of the dataset is to provide a setting in which the community can evaluate rich long-term video understanding.

Q. Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

A. The VISOR annotations were the joint work of groups located at the Universities of Bristol, Michigan, and Toronto. The primary contributors are Ahmad Darkhalil (PhD candidate), Dandan Shan (PhD candidate), Bin Zhu (Postdoctoral researcher), Jian Ma (PhD candidate), Amlan Kar (PhD candidate), Richard Higgins (PhD candidate), Sanja Fidler (Faculty), David Fouhey (Faculty), and Dima Damen (Faculty).

Q. Who funded the creation of the dataset? *If there is an associated grant, please provide the name of the grantor and the grant name and number.*

A. The dataset was funded in multiple parts. The segmentation annotations themselves were funded by charitable unrestricted donation from Procter and Gamble as well as charitable unrestricted donation from DeepMind.

Support for the dataset creation (i.e., student and PI support) was funded as follows: (a) Research at the University of Bristol is supported by UKRI Engineering and Physical Sciences Research Council (EPSRC) Doctoral Training Program (DTP), EPSRC Fellowship UMPIRE (EP/T004991/1) and EPSRC Program Grant Visual AI (EP/T028572/1); (b) Research at the University of Michigan is based upon work supported by the National Science Foundation under Grant No. 2006619; (c) Research at the University of Toronto is in part sponsored by NSERC as well as support through the Canada CIFAR AI Chair program.

(EK) A. The EPIC-KITCHENS videos were funded by a charitable unrestricted donation from Nokia Technologies and the Jean Golding Institute at the University of Bristol.

Q. Any other comments?

A. No

Composition

Q. What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? *Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.*

A. The VISOR dataset contains segments (i.e., a collection of pixel masks) and relations between these segments. These relations consist of: (a) both open-vocabulary and mapped closed-vocabulary semantic entity names that link the segments of the same category; (b) relationships that link each hand in the dataset to at most one object that the hand is in contact with; and (c) long-term relationships between objects and the container from which they emerge in a previous frame.

(EK) A. VISOR is built upon the EPIC-KTICHENS-100 dataset. EPIC-KITCHENS-100 contains 700 variable-length videos along with extensive metadata and labelling, and VISOR has been annotated on frames from this dataset.

Q. How many instances are there in total (of each type, if appropriate)?

A. VISOR contains: (a) ≈271.6K manually segmented semantic masks covering 257 object classes and ≈9.9M automatically obtained dense masks; (b) 67K hand-object relations; (c) exhaustive labels for each segment indicating whether workers believe it to be exhaustive; and (d) 222 instances of long-term tracking.

Q. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? *If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).*

A. VISOR is annotated on sparse frames at a rate of approximately 2 per EPIC-KITCHENS action. This subselection is needed because it is not possible to annotate every frame in the dataset.

Q. What data does each instance consist of? *"Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.*

A. The VISOR dataset consists of multiple types of instance. The segments consist of: (a) a frame-level mask, (b) short-term track ID linking masks within a subsequence, (c) open-vocabulary semantic entity name, (d) closed-vocabulary grouping into one category, and (e) exhaustive flag on whether the mask covers all instances of that category in the image. For **hands** you additionally have (f) hand side, (g) hand contact state and (h) ID of mask in-contact with the hand if present.

Q. Is there a label or target associated with each instance? *If so, please provide a description.*

A. The VISOR dataset consists of labels. See the description of the instances above. The entity classes are also mapped into macro-classes for a one-level hierarchical grouping. This is inherited from EPIC-KITCHENS-100.

Q. Is any information missing from individual instances? *If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.*

A. Yes, we flag in-exhaustively annotated instances. We also only segment active objects in the frame and do not segment background objects. This not only highlights active objects but also avoids expensive annotations for objects lying around and of no relevance to the ongoing action.

Q. Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? *If so, please describe how these relationships are made explicit.*

A. The segments of the dataset are linked in a variety of ways: (a) segments within the same subsequence share a unique ID, (b) segments that have the same name are linked; (c) hands that are in contact with objects are linked; (d) a number of segments are linked to the segments from which they emerge earlier in the video.

Q. Are there recommended data splits (e.g., training, development/validation, testing)? *If so, please provide a description of these splits, explaining the rationale behind them*

A. Yes, we provide splits that are detailed in Sec of the paper. We first use the Test split from EPIC-KITCHENS-100 as the source of our test set. The action annotations in this split are hidden and only available by submitting to the evaluation server. This ensures our test annotations for VISOR are also suitable to use for a formal challenge. We provide a new train/val split from the train and val videos of EPIC-KITCHENS-100. This focuses on ensuring: (a) a number of unseen kitchens are available in Val to assess generality in the same way as Test; (b) some zero-shot classes exist; (c) an 80-20 split of masks is roughly selected per seen kitchen. We use the same Train/Val/Test splits for all the VISOR challenges.

Q. Are there any errors, sources of noise, or redundancies in the dataset? *If so, please provide a description.*

A. Noise and errors are inevitable in datasets. The most likely source of errors is incorrect labels or fundamental ambiguities in labeling. However, the VISOR dataset collection process has multiple quality assurance steps that aims to substantially reduce the prevalence of noise and errors.

Q. Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? *If it links to or relies on external resources, a) are there*

guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.

A. The dataset relies on EPIC-KITCHENS-100. (a) EPIC-KITCHENS-100 is available via the data.bris service, which provides for long-term preservation of the dataset even in the case that PIs move institution. (b) We used this version of the data, but to prevent issues with consistent extraction of frames from videos, we will provide our copies of the frames that were used. (c) While there are licensing restrictions (non-commercial use), these licensing restrictions also apply to VISOR.

EPIC-KITCHENS-100 is accessible via this data.bris link.

Q. Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor–patient confidentiality, data that includes the content of individuals' non-public communications)? *If so, please provide a description.*

A. No. VISOR only contains segments and relations between them.

(EK) A. We do not believe that EPIC-KITCHENS-100 contains confidential information. Participants reviewed their footage before release. It is stored in a GDPR-compliant server.

Q. Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? *If so, please describe why*

A. No. VISOR only contains segments.

(EK) A. We do not believe so. The data shows samples of cooking and other daily kitchen activities.

Q. Does the dataset relate to people? *If not, you may skip the remaining questions in this section.*

A. The VISOR dataset relates to people since it consists of annotations for egocentric data collected by people doing daily activities in the kitchen. The footage and annotations are otherwise anonmous.

Q. Does the dataset identify any subpopulations (e.g., by age, gender)? *If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.*

A. No. The participants from the base EPIC-KITCHENS dataset are anonymous.

Q. Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? *If so, please describe how*

A. VISOR contains only segments. We do not believe this is possible from VISOR data alone

(EK) A. It is possible not possible to identify individuals in the dataset. The data has been stripped of information that would make this easy. The consent forms linking participant IDs to their identities are not public and stored securely at the University of Bristol.

Q. Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? *If so, please provide a description.*

A. VISOR contains only segments. We do not believe this is possible from VISOR data alone.

(EK) A. The EPIC-KITCHENS data may reveal information about racial or ethic origin, sex, and location due to the participants visible hands and kitchen contents. We do not believe that the EPIC-KITCHENS dataset contains sensitive data. Two factors also make this less likely: participants in the dataset are anonymous, and participants collected the footage themselves and reviewed it before its inclusion in the dataset.

Collection Process

Q. How was the data associated with each instance acquired? *Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.*

A. A full description appears in the associated paper and its appendix. However, briefly:

1. *Frame and Entity Identification.* Frames and entities in the each frame to be labelled were identified via a mix of rules, crowdsoucing, and student work.
2. *Pixel Labelling.* A freelancer annotator segmented the entity in each frame; each video was annotated by a single annotator who had the ability to move back and forth through time.
3. *Correction of Pixel Labelling.* These segments were checked for consistency by researchers in our lab.
4. *Extra annotation (Exhaustive Labels, Hand-Object Relations).* The segments were annotated with extra information (e.g., exhaustive annotations, hand-object relations) by a crowdsourcing company.

Additionally, the VISOR dense annotations were created by a deep learning model.

Q. What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? *How were these mechanisms or procedures validated?*

A. A full description appears in the associated paper and its appendix. However, briefly:

1. *Frame and Entity Identification.* This stage was done by a mix of the Amazon Mechanical Turk platform and a custom interface created for this project.
2. *Pixel Labelling.* This stage was done via the TOronto Annotation Suite (TORAS).
3. *Correction of Pixel Labelling.* This stage was done via a custom interface created for this project.
4. *Extra annotation (Exhaustive Labels, Hand-Object Relations).* This stage was done via The Hive's platform.

Q. If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

A. There is not a larger set of labels from which VISOR is subsampled. However, VISOR is annotated on a subset of the frames of the EPIC-KITCHENS-100 dataset. This subset was selected from with the goal of avoiding motion blur in the annotated frames.

Q. Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

A. The annotation process for VISOR consists of multiple stages.

1. *Frame and Entity Identification.* Crowdworkers from Amazon Mechanical Turk performed this task with graduate students involved in the project providing quality control. We paid $11.25 per thousand actions annotated. We provide these as HITs of 16 consecutive actions each. Graduate students did this as part of their normal responsibilities on the project.
2. *Pixel Labelling.* Contractors from Upwork performed the pixel annotation. Contractors were paid $6-9 an hour based on experience with higher rates reserved for experienced annotators who also performed quality assurance checks.
3. *Correction of Pixel Labelling.* Researchers involved in the project performed this task alongside other members from the Machine Learning and Computer Vision group at the University of Bristol. These volunteered for the task and are acknowledged in the paper.
4. *Extra annotation (Exhaustive Labels, Hand-Object Relations).* We paid Hive (thehive.ai) to obtain annotations for these annotations. We paid Hive $20 per thousand tasks annotated.

Q. Over what timeframe was the data collected? *Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created.*

A. The annotations were collected over the a period of 22 months with the bulk of the collection done in the year from June 2021 – June 2022.

(EK) A. The collection time of the underlying EPIC-KITCHENS data and annotations spanned Apr 2017-July 2020.

Q. Were any ethical review processes conducted (e.g., by an institutional review board)? *If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation*

A. VISOR is new annotations, rather than new data.

(EK) A. EPIC-KITCHENS was collected with University of Bristol faculty ethics approval. These application is held at the university of Bristol. Participant consent form is available here.

Q. Does the dataset relate to people? *If not, you may skip the remaining questions in this section.*

A. Yes. VISOR consists of annotations for videos showing people performing daily activities in their kitchen.

Q. Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)

A. Not applicable to VISOR.

(EK) A. Yes. This data was collected directly by and with individuals in question.

Q. Were the individuals in question notified about the data collection? *If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.*

A. Not applicable to VISOR.

(EK) A. Yes. Since the data was directly collected by the participants, the participants were aware of the data collection process. All participants were given the opportunity to ask questions before participating, and they could withdraw at any time without giving a reason. Participants consented to the process and watched their footage. All participants were volunteers and were not compensated.

Q. Did the individuals in question consent to the collection and use of their data? *If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.*

A. Not applicable to VISOR.

(EK) A. Yes. The participants consented to data the collection and use of their data. In particular, they reviewed their footage before its use in the dataset.

Q. If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? *If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).*

A. Not applicable to VISOR.

(EK) A. Participants were able to withdraw from the process at any point until the data was published by DOI. At the moment, participants are unable to withdraw their data.

Q. Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? *If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.*

(EK) A. The university of Bristol faculty ethics committee have reviewed the protocol, and approved the dataset. They checked any potential impact and as the data is anonymous no further actions were deemed as needed.

Q. Any other comments?

A. No

Preprocessing/cleaning/labeling

Q. Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? *If so, please provide a description. If not, you may skip the remainder of the questions in this section.*

A. The data consists of labels, so naturally labelling was done. There were multiple stages of cleaning of the data. This cleaning, however, aimed to fix inconsistencies in labelling (e.g., correcting typos).

Q. Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? *If so, please provide a link or other access point to the "raw" data.*

A. There is no raw data besides incorrect earlier versions of the data, such as annotations before typographic errors were fixed.

Q. Is the software used to preprocess/clean/label the instances available? *If so, please provide a link or other access point.*

A. Primarily no. Some of the software is proprietary (e.g., the TORAS labelling software); some of the software is one-off script that are not of interest due their simplicity and non-general purpose nature.

Q. Any other comments?

A. No

Uses

Q. Has the dataset been used for any tasks already? *If so, please provide a description.*

A. We use VISOR in conjunction with EPIC-KITCHENS-100 to solve three challenges: (a) video object segmentation; (b) hand-object segmentation; (c) and a *Where Did This Come From?* challenge. These are challenges are documented in our paper.

Q. Is there a repository that links to any or all papers or systems that use the dataset? *If so, please provide a link or other access point*

A. No

Q. What (other) tasks could the dataset be used for?

A. We anticipate that the data will be useful for many different long-term pixel-grounded video understanding tasks.

Q. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? *For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?*

A. While the EPIC-KITCHENS videos were collected in 4 countries by participants from 10 nationalities, it is in no way representative of all kitchen-based activities globally, or even within the recorded countries. Models trained on this dataset are thus expected to be exploratory, for research and investigation purposes. Anticipating unintended consequences of data is difficult. Here are some potential issues that we see in the data.

1. The frames that are annotated are selected to be easy to annotate, and therefore may have little motion blur. Models may require motion blur augmentation in order to generalise to all frames.
2. Due to the collection of data collection process, the data shows fewer unique individuals compared to e.g., Internet data. This may make it harder to generalise.
3. The verb and noun classes in no way cover all actions and objects present, even in the kitchens recorded.
4. The dataset is naturally-long tailed. Accordingly the models will be biased to better recognise head classes.

Q. Are there tasks for which the dataset should not be used? *If so, please provide a description.*

A. VISOR is available for non-commercial research purposes only. Accordingly, it should not be used for commercial purposes. A commercial license can be acquired through negotiation with the University of Bristol.

Q. Any other comments?

A. No

Distribution

Q. Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? *If so, please provide a description.*

A. Yes. The VISOR will be publicly available for non-commercial research purposes, just like its base data, EPIC-KITCHENS-100.

Q. How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? *Does the dataset have a digital object identifier (DOI)?*

A. The dataset will be released via the University of Bristol data.bris data repository. This repository assigns unique DOIs upon deposit.

Q. When will the dataset be distributed?

A. The dataset will be publicly released on (or before) 1 August 2022.

Q. Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? *If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.*

A. VISOR will be released under a Creative Commons BY-NC 4.0 license, which restricts commercial use of the data.

Q. Have any third parties imposed IP-based or other restrictions on the data associated with the instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.*

A. No third parties have imposed restrictions on VISOR.

Q. Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.*

A. No. There are no restrictions beyond following the non-commercial license.

Q. Any other comments?

A. No

Maintenance

Q. Who will be supporting/hosting/maintaining the dataset?

A. The dataset will be released via the University of Bristol data.bris data repository. This enables long-term preservation of the data even if the PIs change institutions.

Q. How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

A. The creators of the dataset are listed in this document and can be contacted via email.

Q. Is there an erratum? *If so, please provide a link or other access point.*

A. Not at the time of release. If there are errata or updates, we will provide them on the dataset website.

Q. Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? *If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?*

A. We do not have concrete plans as of yet; we will announce any updates on the dataset website.

Q. If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? *If so, please describe these limits and explain how they will be enforced*

A. This does not apply to VISOR.

(EK) A. There are no limits on the retention of data.

Q. Will older versions of the dataset continue to be supported/hosted/maintained? *If so, please describe how. If not, please describe how its obsolescence will be communicated to users.*

A. Yes. The version released is fixed by the DOI https://doi.org/10.5523/bris.2v6cgv1x04ol22qp9rm9x2j6a7 and will not be changed. The university of Bristol is committed to storage and maintenance of the dataset for 20 years.

Q. If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? *If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.*

A. Users are free to extend the dataset on their own and create derivative works, so long as they follow the license agreement. This was also the case for EPIC-KITCHENS-100. There is, however, no official mechanism to integrate user contributions into a new version of the dataset.

Q. Any other comments?

A. No