UNIVERSITY OF BRISTOL

Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100

Dima Damen[1] · Hazel Doughty[1,3] · Giovanni Maria Farinella[2] · Antonino Furnari[2] · Evangelos Kazakos[1] · Jian Ma[1] · Davide Moltisanti[1,4] · Jonathan Munro[1] · Toby Perrett[1] · Will Price[1] · Michael Wray[1]

Abstract
This paper introduces the pipeline to extend the largest dataset in egocentric vision, EPIC-KITCHENS. The effort culminates in EPIC-KITCHENS-100, a collection of 100 hours, 20M frames, 90K actions in 700 variable-length videos, capturing long-term unscripted activities in 45 environments, using head-mounted cameras. Compared to its previous version (Damen in Scaling egocentric vision: ECCV, 2018), EPIC-KITCHENS-100 has been annotated using a novel pipeline that allows denser (54% more actions per minute) and more complete annotations of fine-grained actions (+128% more action segments). This collection enables new challenges such as action detection and evaluating the "test of time"—i.e. whether models trained on data collected in 2018 can generalise to new footage collected two years later. The dataset is aligned with 6 challenges: action recognition (full and weak supervision), action detection, action anticipation, cross-modal retrieval (from captions), as well as unsupervised domain adaptation for action recognition. For each challenge, we define the task, provide baselines and evaluation metrics.

Keywords Video dataset · Egocentric vision · First-person vision · Action understanding · Multi-benchmark large-scale dataset · Annotation quality

✉ Dima Damen
Dima.Damen@bristol.ac.uk
https://dimadamen.github.io/

Hazel Doughty
hazel.doughty@uva.nl
https://hazeldoughty.github.io

Giovanni Maria Farinella
gfarinella@dmi.unict.it
https://www.dmi.unict.it/farinella/

Antonino Furnari
furnari@dmi.unict.it
http://www.antoninofurnari.it

Evangelos Kazakos
evangelos.kazakos@bristol.ac.uk
https://ekazakos.github.io/

Jian Ma
jian.ma@bristol.ac.uk
https://majian8.github.io/jianma/

Davide Moltisanti
davide.moltisanti@ntu.edu.sg
https://www.davidemoltisanti.com/research/

Jonathan Munro
jonathan.munro@bristol.ac.uk
https://jonmun.github.io

Toby Perrett
toby.Perrett@bristol.ac.uk
https://tobyperrett.github.io

Will Price
will.price@bristol.ac.uk
https://www.willprice.dev/research/

Michael Wray
michael.wray@bristol.ac.uk
https://mwray.github.io

[1] University of Bristol, Bristol, UK

[2] University of Catania, Catania, Italy

[3] Present Address: University of Amsterdam, Amsterdam, Netherlands

[4] Present Address: NTU, Singapore, Singapore

1 Introduction and Related Datasets

Since the dawn of machine learning for computer vision, datasets have been curated to train models, for single tasks from classification (Deng et al. 2009; Carreira and Zisserman 2017) to detection (Lin et al. 2014; Gu et al. 2018), captioning (Karpathy and Fei-Fei 2015; Xu et al. 2016) and segmentation (Zhou et al. 2017; Perazzi et al. 2016). Increasingly, datasets have been used for novel tasks, through pre-training (He et al. 2019; Mettes et al. 2016), self-supervision (Noroozi and Favaro 2016; Vondrick et al. 2018) or additional annotations (Gupta and Malik 2016; Heilbron et al. 2018). However, task adaptation demonstrates that models overfit to the data and annotations (Zhai et al. 2019; Moltisanti et al. 2017).

Alternatively, one dataset can be enriched with multiple annotations and tasks, aimed towards learning intermediate representations through downstream and multi-task learning on the same input. This has been recently achieved for autonomous driving (Zhou et al. 2019; Geiger et al. 2012; Cordts et al. 2016; Neuhold et al. 2017; Yu et al. 2018; Huang et al. 2018; Caesar et al. 2019; Yogamani et al. 2019) and scene understanding (Zamir et al. 2018; Silberman et al. 2012). For example, Zamir et al. (2018) contains 26 tasks ranging from edge detection to vanishing point estimation and scene classification.

In comparison, the number of tasks proposed for action and activity understanding datasets (Damen et al. 2018; Gu et al. 2018; Heilbron et al. 2015; Rohrbach et al. 2015; Zhou et al. 2017; Rohrbach et al. 2012) remains modest. Often, this is limited by the source of videos in these datasets. YouTube (Heilbron et al. 2015; Zhou et al. 2017) and movies (Gu et al. 2018; Rohrbach et al. 2015) typically contain curated videos, with edited shots. However, attempts to define multiple challenges for these datasets have been exemplary. ActivityNet (Heilbron et al. 2015) is the most popular video challenge, evaluated for localisation, dense captioning (Krishna et al. 2017) and object detection (Zhou et al. 2019). Similarly, AVA (Gu et al. 2018) has challenges on action localisation and active speaker detection (Roth et al. 2019).

Several leading egocentric datasets (Pirsiavash and Ramanan 2012; Damen et al. 2014; Fathi et al. 2012; De La Torre et al. 2008; Li et al. 2015) showcased the unique perspective and potential of first-person views for action recognition, particularly hand-object interactions. In 2018, the introduction of the largest-scale dataset EPIC-KITCHENS (Damen et al. 2018) has transformed egocentric vision, not only due to its size, but also the unscripted nature of its collection and the scalable nature of the collection pipeline. In this paper, we present EPIC-KITCHENS-100, a substantial extension which brings the total footage to **100** hours, capturing diverse unscripted and unedited object inter-

actions in people's kitchens.[1] As shown in Fig. 1, the actions capture hand object interactions with everyday objects in participants' kitchens. The unscripted nature of the dataset results in naturally unbalanced data, with novel compositions of actions in new environments. While challenging, the dataset is domain-specific (i.e. kitchen-based activities), offering opportunities for engaging domain knowledge. We offer two-level annotations for nouns and verbs in interactions (e.g. "carrot/courgette → vegetable", "put/throw/drop → leave") to utilise such priors.

Importantly, we propose a refined annotation pipeline that results in denser and more complete annotations of actions in untrimmed videos. This pipeline enables various tasks on the same dataset; we demonstrate six in Sect. 4, with baselines and evaluation metrics that focus on understanding fine-grained actions and offer benchmarks which can support research into better modelling of video data.

# 2 Data Collection and Scalable Pipeline

In this section, we detail our collection and annotation effort.

Data Collection We collect additional footage as follows: we contacted participants from EPIC-KITCHENS-55 to record further footage. Of the 32 participants in (Damen et al. 2018), 16 subjects expressed interest in participating. Interestingly, half of these (8 subjects) had moved homes over the past two years. We also recruited 5 additional subjects, increasing the total number of subjects and kitchen environments to 37 and 45 respectively. All participants were asked to collect 2–4 days of their typical kitchen activities, as in (Damen et al. 2018). We collect footage using a head mounted GoPro Hero7 black. This is two generations newer than the camera used in EPIC-KITCHENS-55, with a built-in feature for HyperSmooth video stabilisation. Sample frames are shown in Fig. 1, with selected frames of the same action in returning and changing kitchens.

Annotation Pipeline An overview of the pipeline can be seen in Fig. 2.

(a) Narrator Previously, for EPIC-KITCHENS-55, we used a non-stop narration approach, where each participant narrated their previous action while watching the future actions in the running video. We found this resulted in increased mental load and some actions being missed or misspoken. To improve upon this approach, we take inspiration from Gygli and Ferrari (2019), where objects in images are annotated by pointing and speaking and propose temporal 'pointing' which we refer to as 'pause-and-talk'. By allowing partic-

---

[1] We will refer to the previous edition as EPIC-KITCHENS-55 in reference to the number of hours collected and annotated.
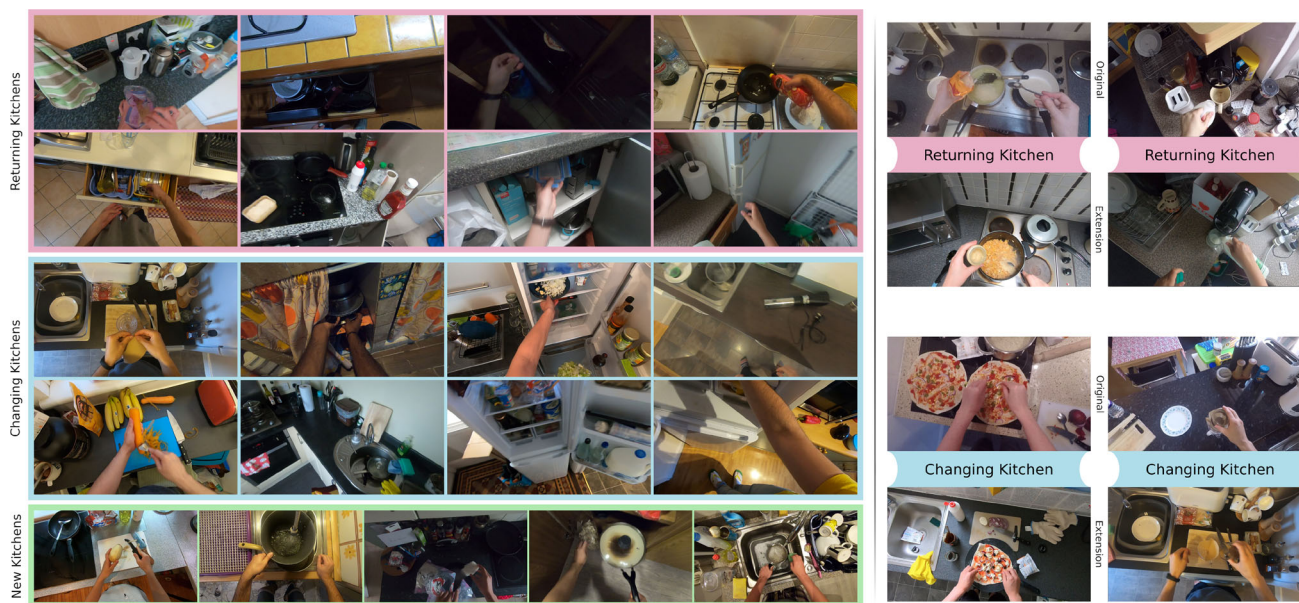
Fig. 1 Left: Frames from EPIC-KITCHENS-100 showcasing returning participants with returning or changing kitchens (top) as well as new participants (bottom). Right: Comparisons between recordings from [1] and newly collected videos, with selected frames showcasing the same action. Note object location differences in 'returning' kitchens (e.g. microwave relocated). We show the same action performed in 'changing' kitchens (e.g. same participant preparing pizza or filtered coffee in a new kitchen)

ipants to pause the video to speak as well as take breaks, we hope to increase accuracy and density of actions, whilst still allowing for a scalable narration approach. We built an interface to facilitate collecting such narrations from the participants (Fig. 2a), which includes a video player, synced with audio recordings.[2] Participants watch the video and press a key to pause while they narrate the action in their native language. As previously observed in Damen et al. (2018), using the native language ensures the narrations use the correct vocabulary in describing the actions. The video restarts on key release. Note that the narrator still watches the video *once*, maintaining the targeted scalability of the annotation pipeline, but removes the mental overload of narrating past actions while watching future actions. This allows for short and overlapping actions to be captured in addition to enabling error correction, as participants can listen to, delete or re-record a narration. Fig. 2 shows an ongoing narration, demonstrating density (ticks on the slider).

(b) Transcriber We perform transcription of audio narrations, followed by translation (if applicable): first, we transcribe all narrations and then translate the unique transcriptions into English using a hired translator for correctness and consistency. The approach we used to transcribe narrations in Damen et al. (2018) had issues where workers failed to understand some audio narrations due to the lack

of any visual information. To mitigate this, we build a new transcriber interface containing three images sampled around the timestamp (Fig. 2b). We find that images increase worker agreement and alleviate issues with homonyms (e.g. 'flower' and 'flour'). Each narration is transcribed into a caption by 3 Amazon Mechanical Turk (AMT) workers using a consensus of 2 or more workers. Transcriptions were automatically rejected if the cosine similarity between the Word2Vec (Mikolov et al. 2013) embeddings was lower than an empirical threshold of 0.9. When AMT workers fail to agree, the correct transcription was selected manually. Captions were then spell checked and substitutions were applied from a curated list of problematic words (e.g. 'hob' and 'hop'), further reducing errors.

(c) Parser We use spaCy (Honnibal and Montani 2017) to parse the transcribed captions into verbs and nouns (Fig. 2c) and manually group these into minimally overlapping classes as we did in our previous work. We reworked this to improve parsing of compound nouns and missing verbs/nouns. Additionally, all annotations (including those we collected previously from EPIC-KITCHENS-55) were re-parsed using the updated pipeline. To cluster the verbs and nouns, we adjust previous clusters to reduce ambiguities between classes. For example, we group 'brush' and 'sweep'

---

[2] Our tool is available at https://github.com/epic-kitchens/epic-kitchens-100-narrator
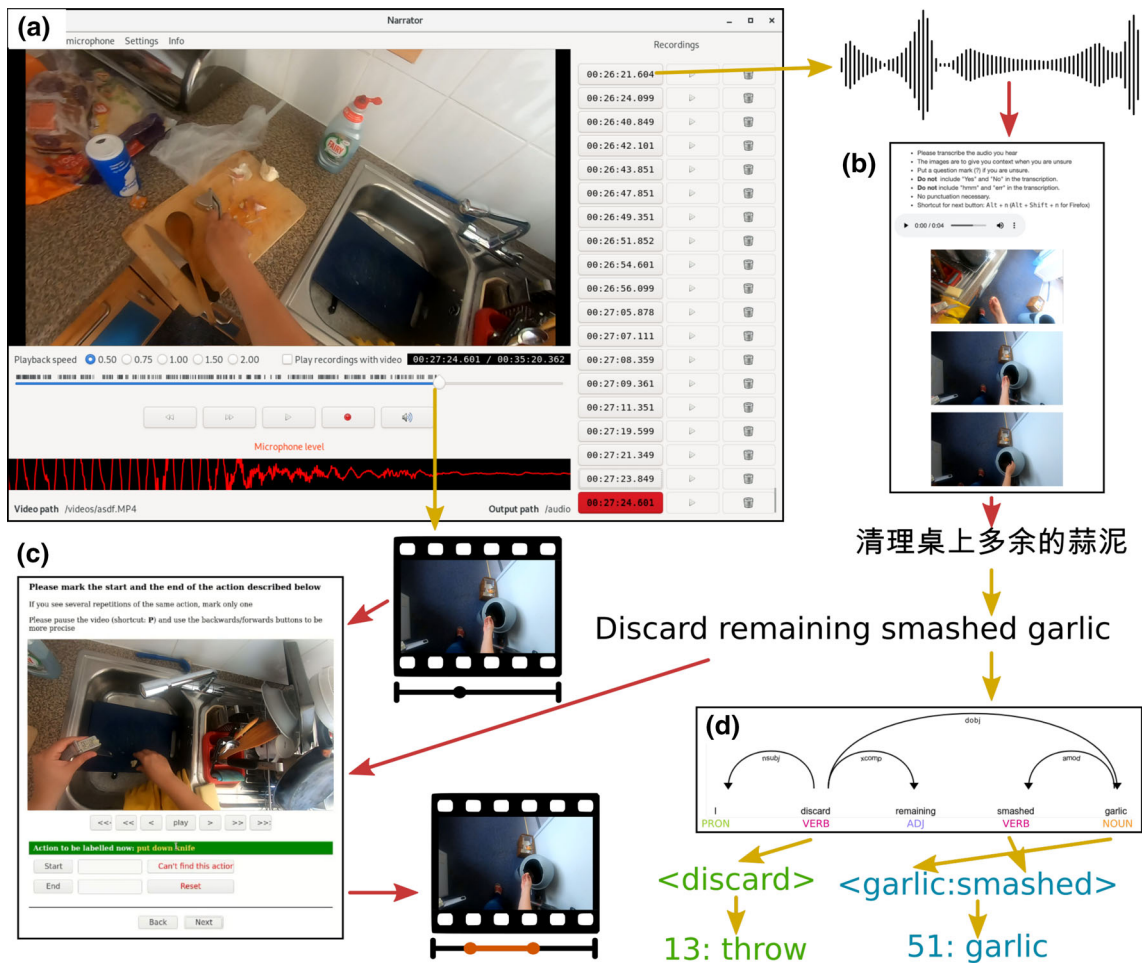
Fig. 2 Annotation pipeline: **a** narrator, **b** transcriber, **c** temporal segment annotator and **d** dependency parser. Red arrows show AMT crowdsourcing of annotations (Color figure online)

into one verb class, and introduce noun classes that did not exist before such as 'lentils'.

(d) Temporal Annotator We built an AMT interface for labelling start/end times of action segments (Fig. 2d). Annotators completed a quick tutorial on annotating temporal bounds before they labelled 10 consecutive actions. To create the bounds of the action segment, we use the same approach as we did previously but increased the number of workers from 4 to 5 to improve quality. Note that in the untrimmed videos there might be consecutive instances of the same action. These will be indicated by repeated narrations. We thus request that annotators mark the temporal bounds of each instance, prompted by the timestamp. This avoids the merging of instances of the same action.

Quality Improvements Our EPIC-KITCHENS-100 scalable pipeline focuses on denser and more accurate annotations. We compare different parts of the pipeline to our previous one in Appendix B. Here, we show improved quality of annotations both numerically and through an example.

Figure 3 (left) compares the narration method we used in Damen et al. (2018) to the new pipeline over several metrics. Our 'pause-and-talk' narrator produces more densely annotated videos; fewer gaps and more labelled frames; actions are shorter; and exhibit higher overlap. The narration timestamps are also closer to the relevant action, with a higher percentage being contained within the action and a smaller distance to remaining timestamps outside the action.

Figure 3 (right) shows two video sections, of equal length, annotated by the same participant, one using non-stop narrations and the other with 'pause-and-talk'. The number of annotated actions increased from 20 to 56, with short actions (such as 'turn on tap') often missed in the previous pipeline. We demonstrate these through two examples. The first shows a missed action of picking up a bag off the floor that had been dropped, and the second shows a missed closing cupboard action. In the sequence from 'pause-and-talk', all actions including closing the cupboard were successfully narrated thanks to our 'pause-and-talk' pipeline. By narrating more actions, the start/end times also become more accurate as it
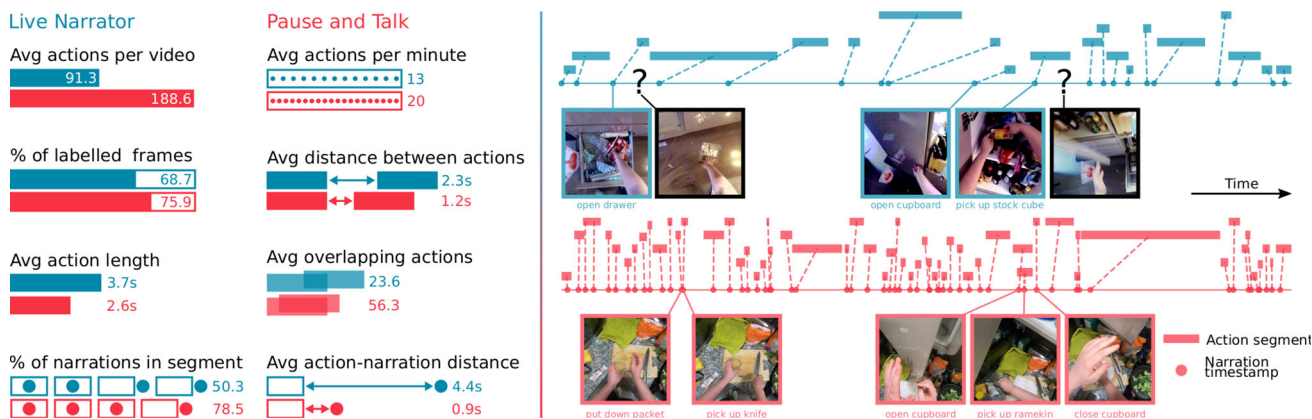
Fig. 3 Comparing non-stop narrations (blue) to 'pause-and-talk' narrations (red). Right: timestamps (dots) and segments (bars) for two sample sequences. 'pause-and-talk' captures all actions including short ones. Black frames depict missed actions (Color figure online)
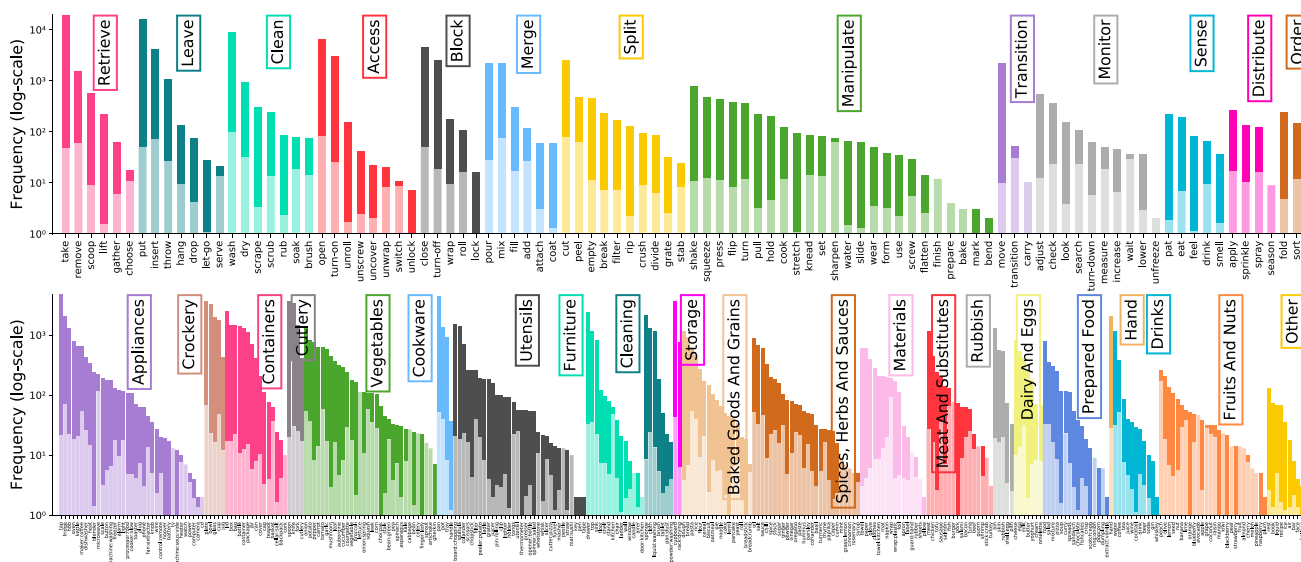
Fig. 4 Frequency of verbs (top) and nouns (bottom), grouped by category. Each bar is *linearly* split: solid represents instances from newly-collected videos and washed-out from original videos

3 Statistics, Scalability and the Test of Time

EPIC-KITCHENS-100 contains 89,977 segments of fine-grained actions annotated from 700 long videos. Footage length amounts to 100 hours. Table 1 lists the general statistics, separating those from the videos collected previously to the newly collected videos. Note that all previous narrations have been re-parsed using the new pipeline (Fig. 2b–d). EPIC-KITCHENS-100 rescales our previous dataset with almost double the length with 1.8x hours and 2.3x action

is more obvious to the AMT annotators what each narration refers to.

segments. Comparisons to other datasets are presented under relevant benchmarks in Section 4.

In Fig. 4 we show the frequency of verb (97) and noun (300) classes in the dataset. These are grouped into categories (13 verb and 21 noun categories), sorted by size. For example, verbs 'wash', 'dry', 'scrape', 'scrub', 'rub', 'soak' and 'brush' are grouped into a *clean* verb category. The plots show a clear long-tail distribution. The contribution of each class from source videos (Damen et al. 2018) and extension are also shown. New verb classes (e.g. 'lock', 'bend') and noun classes (e.g. 'orange' and 'hoover') are only present in the newly-collected videos.

We enrich our dataset with automatic spatial annotations using two models. The first is Mask R-CNN (He et al. 2017) trained on MSCOCO (Lin et al. 2014). The second is hand-
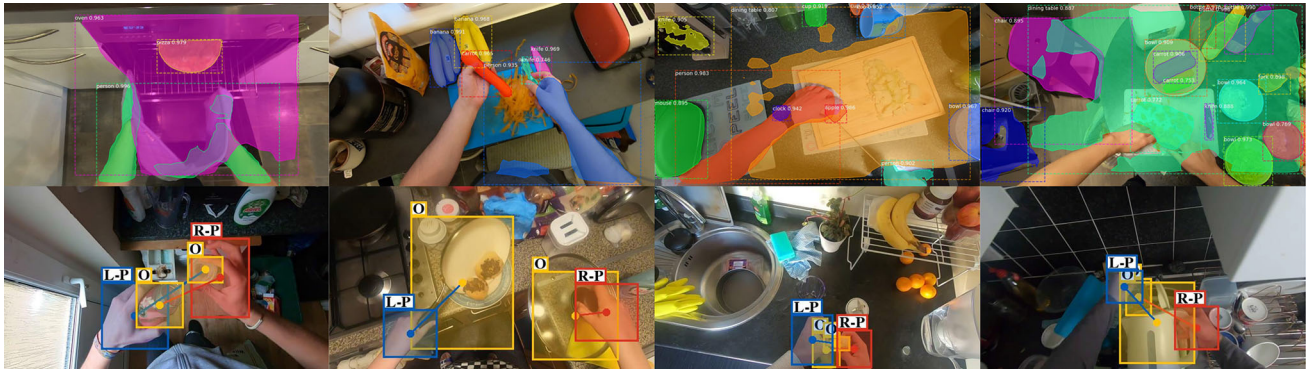
Fig. 5 Top: Sample Mask R-CNN of large objects (col1: oven), hands (labelled person), smaller objects (col2: knife, carrot, banana, col3: clock, toaster, col4: bottle, bowl), incorrect labels of visually ambiguous objects (col3: apple vs onion) and incorrect labels (col3: mouse, col4: chair). Bottom: Sample hand-object detections from Shan et al. (2020). L/R = Left/Right, P = interaction with portable object, O = object. Multiple object interactions are detected (col2: pan and lid, col4: tap and kettle)
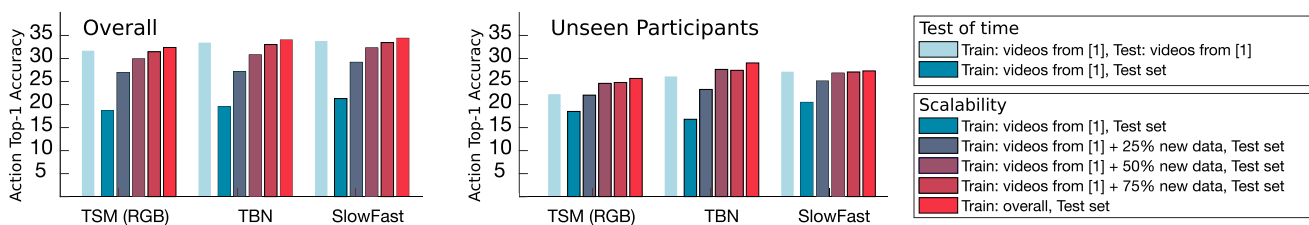
Fig. 6 Test of time and scalability test results

object interactions from Shan et al. (2020), trained on 100K images from YouTube along with 42K images from three egocentric datasets (Damen et al. 2018; Sigurdsson et al. 2018; Li et al. 2015) of which 18K are from our videos (Damen et al. 2018). It detects interacted static and portable objects as an offset to hand detections. Example annotations are shown in Fig. 5, and the number of annotations is given in Table 1. While we do not use these annotations to report results, we believe these 66M masks, 31M hand and 38M object bounding boxes could facilitate future models for spatial (or spatio-temporal) attention.[3]

Splits We split the videos into Train/Val/Test with a ratio of roughly 75/10/15. Each video, with all its action segments, is present in one of the splits, and the Test split contains only newly-collected videos. We use re-parsed videos from the original EPIC-KITCHENS test sets as the new validation set.[4] Our Val/Test splits contain two interesting subsets, which we report on separately:

- *Unseen Participants* Our Val and Test splits contain participants not present in Train: 2 participants in Val, and another 3 participants in Test. These contain 1,065 and 4,110 action segments respectively. This subset helps evaluate the generalisability of the models across the various benchmarks.
- *Tail Classes* We define these (for verbs and nouns) to be the set of smallest classes whose instances account for 20% of the total number of instances in training. A tail action class contains either a tail verb class or a tail noun class. These are 86/228/3,729 verb/noun/action classes.

Scalability and the Test of Time As we rescale EPIC-KITCHENS with additional videos, we carry out two investigations: (a) how models trained on videos from Damen et al. (2018) perform on videos collected two years later, and (b) how models' performance scales with additional annotated data. We call these the *test of time* and the *scalability* tests respectively.

Figure 6 includes results for both investigations, evaluated on the task of action recognition (definition and models from Sect. 4.1). We separate overall results (left) from unseen participants (right). For all models, comparing the first two bars demonstrates that models trained solely on videos from

---

[3] Correctness of bounding boxes for hands and objects has been evaluated by Shan et al. (2020)—see acknowledgements. Performance of R-CNN masks has not been quantitatively evaluated and these are error-prone.

[4] We no longer split the test set into seen and unseen kitchens, but instead report on relevant evaluation metrics for each challenge.

Damen et al. (2018) do not withstand the *test of time*. For the same model, performance drops significantly when new data is evaluated. This highlights a potential domain gap, which we discuss next. We assess *scalability* by gradually adding new data in training. Results demonstrate a significant improvement, albeit saturating when 50% of new data is added, particularly for unseen participants. This highlights the need for better models and more diverse data rather than merely more data. This can be particularly observed as the unseen participants data benefits even less when scaling. We tackle the gap to new environments and participants next.

Unravelling the Domain Gap As defined in the early work on speech recognition (Ueberla 1997), "A domain D is a (often infinite) set of samples such that each sample satisfies a property P_D". A domain gap is present when at least one property differs between the samples of two domains. Domain gaps have been a frequent source of frustration for a wide range of learning tasks, where models are trained on samples from one domain, and thus under-perform when deployed in a different domain. This is also known as sample-selection bias (Heckman 1979). Sampling bias is a common cause for a domain gap between datasets, which cannot easily be removed during dataset collection, as noted in (Torralba and Efros 2011). The most obvious domain gaps stem from changes in locations (Oberdiek et al. 2020), viewpoints (Zhai et al. 2017), labels (Hsu et al. 2020) and participants (Stein and McKenna 2013). However, there are often more subtle causes, such as differences in capture methodology (Saenko et al. 2010) or due to changes in objects, environments and actions over time.

The concept of a compound domain gap has recently been introduced in Liu et al. (2020), where the target domain is a compound of multiple domains without domain labels. As stated by Liu et al. (2020), this is a more realistic scenario resulting from unconstrained data collection. In EPIC-KITCHENS-100, each video in the extension offers a compound domain gap due to changes in one or more of the following properties:

– Hardware and capturing as in Saenko et al. (2010); Gong et al. (2012). Extended footage uses a newer camera model with onboard video stabilisation.
– Locations as in Oberdiek et al. (2020). As indicated in Section 2, eight subjects have moved home resulting in changing surroundings but keeping the appearance of many objects and tools. Additionally, unseen participants capture footage in new environments where the appearance of objects and surroundings differ.
– Participants as in Stein and McKenna (2013). Hand appearance and individual behaviours exist in the extension which are not in the original footage.

Table 1 Statistics of EPIC-KITCHENS-100 and its Train/Val/Test splits

	Hours	Videos	Action Seg.	Unique Narr.	Verb Cls.	Noun Cls.	Action Cls.	Object Masks	Hand BB	Int. Obj
Source										
Videos from Damen et al. (2018)	54.6	432	39,432	11,423	93	272	2,747	35,682,398	18,234,678	22,156,746
Extension	45.4	268	50,547	11,236	91	266	2,900	29,987,598	12,999,913	16,043,057
Overall	**100.0**	**700**	**89,977**	**20,580***	**97**	**300**	**4,053**	**65,669,996**	**31,234,591**	**38,199,803**
Splits										
Train	74.7	495	67,217	15,968	97	289	3,568	48,896,723	23,186,294	28,190,446
Val	13.2	138	9,668	3,835	78	211	1,352	8,714,871	4,462,472	5,513,884
Test	12.1	67	13,092	4,324	84	207	1,487	8,058,402	3,585,825	4,495,473

*There is overlap between the unique narrations of Damen et al. (2018) and extension, hence overall does not sum up across sources
Bold signifies overall released version statistics

- Short-term temporal offsets as in Wulfmeier et al. (2018), where time-of-day can affect scene lighting, and some background objects change position (e.g. on the counter for one video, put away in a cupboard for a later video).
- Long-term temporal offsets as in Carlevaris-Bianco et al. (2016); Maddern et al. (2017). EPIC-KITCHENS-100 is filmed 2 years after EPIC-KITCHENS-55. In the same environment, changes such as wear and tear, new objects and different object positions are observed (see Fig. 1 right). Participant behaviour can also change over time.

While we have domain labels for some of these properties (e.g. recording camera, location, time-of-day and participant ID), other property changes can vary between samples, without associated labels. It is particularly difficult to associate labels with changes in behaviour or object appearances, for example. We publish these properties with the dataset when present. Importantly, we explore this compound domain gap, without using property labels, using a new challenge on unsupervised adaptation for action recognition (Sect. 4.5).

4 Challenges and Baselines

In this section, we define 6 challenges on our dataset, two modified from Damen et al. (2018), namely action recognition (Sect. 4.1) and anticipation (Sect. 4.4). We introduce four new challenges: weakly-supervised action recognition (Sect. 4.2), action detection (Sect. 4.3), unsupervised domain adaptation for action recognition (Sect. 4.5) and action retrieval (Sect. 4.6). While many works have addressed one or more of these challenges, they are typically explored using different datasets. Our annotation pipeline (from captions and single timestamps to segments and classes—Fig. 2) can be used to define multiple challenges, potentially jointly. In this section, we only scratch the surface by reporting on each challenge independently. For readability, we include all implementation details in Appendix C, and we published all our baseline models and evaluation scripts.

4.1 Action Recognition

Definition As in Damen et al. (2018), we consider a video segment (t_s, t_e) as the start and end frames in a video. We aim to predict $(\hat{v}, \hat{n}, \hat{a})$ as the verb/noun/action classes of the action in this segment. We consider overlapping segments independently.

Related Datasets Several datasets have been collected to focus on action recognition, from Soomro et al. (2012), Kuehne et al. (2011) to recent large-scale ones (Gu et al. 2018; Kay et al. 2017; Monfort et al. 2020; Goyal et al. 2017; Zhao et al. 2019; Sigurdsson et al. 2016), all offering

a challenge with a held-out test set. In Table 2, we compare EPIC-KITCHENS-100 to these non-egocentric datasets across a range of facets. Ours is the only dataset of unscripted activities, of comparable size to those collected from scripted or curated (YouTube) videos.

Evaluation Metrics We report Top-1/5 Accuracy on Val and Test sets.

Baselines and Results In Table 3, we report results of five state-of-the-art recognition models (Wang et al. 2016; Zhou et al. 2018; Kazakos et al. 2019; Lin et al. 2019; Feichtenhofer et al. 2019) in addition to a random chance baseline. We use the Train set to report on Val, optimising hyper-parameters. We then fix these, and train on both the Train and Val sets in order to report on the Test set. Figure 7 shows success and failure examples, using examples from the Val set.

4.2 Weakly-Supervised Action Recognition

Definition As in Sect. 4.1, the goal is to recognise the action, *i.e.* predict $(\hat{v}, \hat{n}, \hat{a})$, in a *trimmed* action segment during *testing*. Distinctly, we use single timestamps instead of temporal boundaries during *training*. Let $\mathcal{A} = (A_i)_{i=1}^N$ be the action instances contained in an untrimmed training video, each $A_i = (t, v, n, a)$ is labelled with only one timestamp t roughly located around the action, along with verb/noun classes. We utilise the narration timestamps from our collection pipeline as t.

Related Datasets and Types of Supervision Previous weakly-supervised approaches utilised video-level or transcript supervision, where the set (Wang et al. 2017; Singh and Lee 2017; Nguyen et al. 2018; Liu et al. 2019; Nguyen et al. 2019; Narayan et al. 2019) or sequence (Bojanowski et al. 2014; Huang et al. 2016; Ding and Xu 2018; Richard et al. 2018; Chang et al. 2019; Li et al. 2019) of actions in the video are used in training, without temporal bounds. Table 4 compares EPIC-KITCHENS-100 to datasets trained with weak-supervision. When considering the number of classes (and instances) per video, EPIC-KITCHENS-100 offers a significant challenge. For example, ActivityNet (Heilbron et al. 2015) videos contain 1 class and 1.5 action instances on average, whereas in EPIC-KITCHENS-100, videos contain 53.2 classes and 128.5 instances. Video-level supervision is only sufficient for datasets with a few classes per video (Heilbron et al. 2015; Jiang et al. 2014), while transcript supervision (Marszalek et al. 2009; Kuehne et al. 2014) expects no overlap between actions. Both types of weak supervision are insufficient in our case.

Alternatively, single-timestamp supervision is gaining popularity due to the scalability and performance balance (Moltisanti et al. 2019; Bearman et al. 2016; Mettes et al. 2016; Chéron et al. 2018). We follow this trend as it fits

Table 2 A comparison of EPIC-KITCHENS-100 against popular action recognition datasets. *a* = Action, *v* = Verb, *n* = Noun, *c* = caption, *ML-a* = Multi-Label Action

Dataset	Year	Type	Hours	Actions per Video	Action Clips	Avg. Action Length	Action Classes	Task	Metrics
EPIC-KITCHENS-100	2020	Unscripted	100	128.5	90k	3.1 ± 5.4	4,053	a, v, n	Top-1/5 Acc.
Kinetics-700 (Kay et al. 2017; Carreira et al. 2019)	2019	YouTube	1806	1	650k	10.0 ± 0.0	700	a	Top-1/5 Err.
Multi-Moments in Time (Monfort et al. 2020)	2019	YouTube	850	1	1M	3.0 ± 0.0	313	ML-a	mAP
Something-Something V2 (Goyal et al. 2017; Mahdisoltani et al. 2018)	2018	Scripted	234	1	220k	3.8 ± 1.1	174	a, n, c	Top-1/5 Acc.
AVA (Roth et al. 2019)	2018	Film	108	1380	410k	2.7 ± 3.5	80	ML-a	mAP
HACS (Zhao et al. 2019)	2017	YouTube	861	2	1.5M	2.0 ± 0.0	200	a	mAP
Charades (Sigurdsson et al. 2016)	2016	Scripted	81	6.8	67k	12.8 ± 9.3	157	ML-a	mAP

Table 3 Action recognition results on Val (using Train) and Test (using Train+Val)

Split	Baseline	Overall Top-1 Accuracy (%)			Top-5 Accuracy (%)			Unseen Participants Top-1 Accuracy (%)			Tail Classes Top-1 Accuracy (%)		
		Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	Chance	10.42	01.70	00.51	38.43	08.14	02.54	10.59	01.88	00.57	01.13	00.31	00.10
	TSN (Wang et al. 2016)	60.18	46.03	33.19	89.59	72.90	55.13	47.42	38.03	23.47	30.45	19.37	13.88
	TRN (Zhou et al. 2018)	65.88	45.43	35.34	90.42	71.88	56.74	55.96	37.75	27.70	34.66	17.58	14.07
	TBN (Kazakos et al. 2019)	66.00	47.23	36.72	90.46	73.76	57.66	59.44	38.22	29.48	39.09	24.84	19.13
	TSM (Lin et al. 2019)	67.86	49.01	38.27	90.98	74.97	60.41	58.69	39.62	29.48	36.59	23.37	17.62
	SlowFast (Feichtenhofer et al. 2019)	65.56	50.02	38.54	90.00	75.62	58.60	56.43	41.50	29.67	36.19	23.26	18.81
Test	Chance	10.68	01.79	00.55	37.71	08.35	02.69	09.37	01.90	00.59	00.97	00.39	00.12
	TSN (Wang et al. 2016)	59.03	46.78	33.57	87.55	72.10	53.89	53.11	42.02	27.37	26.23	14.73	11.43
	TRN (Zhou et al. 2018)	63.28	46.16	35.28	88.33	72.32	55.26	57.54	41.36	29.68	28.17	13.98	12.18
	TBN (Kazakos et al. 2019)	62.72	47.59	35.48	88.77	73.08	56.34	56.69	43.65	29.27	30.97	19.52	14.10
	TSM (Lin et al. 2019)	65.32	47.80	37.39	89.16	73.95	57.89	59.68	42.51	30.61	30.03	16.96	13.45
	SlowFast (Feichtenhofer et al. 2019)	63.79	48.55	36.81	88.84	74.49	56.39	57.66	42.55	29.27	29.65	17.11	13.45

Table 4 Characteristics of popular datasets related to our challenges: weakly-supervised action recognition (WS), anticipation (Ant.) and detection (Det.)

Dataset	WS	Ant.	Det.	Classes	Instances	Hours	Avg. Action Length	Avg. Instances per Video	Avg. Classes per Video	Labelled Frames	Overlapping Segments
TV Human Interactions (Patron-Perez et al. 2010)	✗	✓	✗	4	300	0.3	3.9s	1.0	1.0	100.0%	0.0%
TV Series (De Geest et al. 2016)	✗	✓	✓	30	6,231	16.0	1.9s	230.7	23.5	N/A	21.9%
THUMOS 14 (Jiang et al. 2014)	✓	✓	✓	101	6,310	30.0	4.2s	15.4	1.1	29.5%	8.3%
Multi-THUMOS (Yeung et al. 2017)	✓	✗	✓	65	38,690	30.0	3.5s	102.1	10.5	78.6%	67.1%
ActivityNet 1.3 (Heilbron et al. 2015)	✓	✗	✓	200	23,064	648.0	49.2s	1.5	1.0	65.3%	1.0%
Charades (Sigurdsson et al. 2016)	✓	✗	✓	157	66,500	81.1	12.8s	6.9	6.8	89.7%	79%
UCF 101-24 (Gorban et al. 2015)	✓	✗	✓	24	4,569	6.2	5.1s	1.4	1.0	82.5%	18.8%
DALY (Weinzaepfel et al. 2016)	✓	✗	✓	10	3,907	31.9	7.7s	7.6	1.1	30.5%	8.4%
Hollywood2 (Marszalek et al. 2009)	✓	✗	✗	16	2,376	21.0	4.3s	2.5	1.9	37.0%	0.0%
Breakfast (Kuehne et al. 2014) (cam01)	✓	✓	✓	8	9,941	77.0	3.0s	38.7	18.5	96.6%	0.4%
50 Salads (Stein and McKenna 2013)	✓	✓	✓	17	899	4.5	36.8s	18.0	16.3	85.9%	3.6%
MPII (Rohrbach et al. 2012)	✓	✓	✓	65	5,609	8.3	11.1m	127.5	24.9	97.3%	0.1%
EGTEA Gaze+ (Li et al. 2015)	✗	✓	✓	106	15,484	28.0	3.7s	180.0	82.8	57.9%	6.1%
EPIC-KITCHENS-100	✓	✓	✓	4,053	89,977	100.0	3.1s	128.5	53.2	71.6%	28.1%

naturally with our narration timestamps collected using 'pause-and-talk'.

Evaluation Metrics We follow the same metrics as in Sect. 4.1.

Baselines and Results We consider two baselines. The first, "Fixed segment", uses a segment of fixed length centred on the timestamp. The second is our previous work (Moltisanti et al. 2019), where sampling distributions, to select training frames from the untrimmed videos, are initialised from single timestamps, and refined based on the classifier's response.[5] Both are trained end-to-end using a TSN backbone (Wang et al. 2016) and results can be seen in Table 5. Moltisanti et al. (2019) improves the fixed segment baseline by 1–3% top-1 accuracy across Val and Test. The fully supervised upper bound is TSN, reported in Table 3. Comparatively, weak supervision performs 11% worse than strong supervision on top-1 action accuracy in Val and Test. Using roughly aligned single timestamps is challenging when actions are short and overlapping. EPIC-KITCHENS-100, with its dense actions, provides an interesting benchmark to develop new models for weak-supervision.

4.3 Action Detection

Definition All other challenges in Section 4 consider a trimmed segment (t_s, t_e) from the test video as input. This assumption is limiting, as labelled start/end times of actions are unlikely to be present for new test videos. In this challenge, we aim to detect and recognise all action instances within an untrimmed video, as in Heilbron et al. (2015). Given a video, we predict the set of all action instances $\hat{\mathcal{A}} = \{\hat{A}_i\}_{i=1}^{M}$, where $\hat{A}_i = (\hat{t}_s, \hat{t}_e, \hat{v}, \hat{n}, \hat{a})$ is an action detection tuple including the predicted start and end times $(\hat{t}_s, \hat{t}_e)$ and the predicted classes $(\hat{v}, \hat{n}, \hat{a})$. During training, we use the set of ground-truth action annotations $\mathcal{A}$. Note that the ground-truth $\mathcal{A}$ and predicted $\hat{\mathcal{A}}$ sets can be of different sizes. This definition is closely related to temporal segmentation (Lea et al. 2017), but segmentation assumes non-overlapping segments and is thus unsuitable for EPIC-KITCHENS-100.

Related Datasets Table 4 compares EPIC-KITCHENS-100 to popular datasets for temporal action detection and segmentation. EPIC-KITCHENS-100 presents the largest challenge, when considering the combined metrics of: average video length, average instances per video and overlapping instances. Compared to datasets with overlapping segments, it has a larger number of instances per video and is also longer

---

[5] The distributions are modelled with a plateau function, initialised with a fixed width and slope, and centred around the annotated timestamp. These are refined from the classification scores iteratively. More details in Moltisanti et al. (2019)

Table 5 Weakly-supervised action recognition results

Split	Baseline	Overall						Unseen Participants			Tail Classes		
		Top-1			Top-5			Top-1			Top-1		
		Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	Fixed segment	44.86	37.97	20.30	84.62	65.41	39.35	37.37	29.20	14.36	25.85	18.89	10.50
	(Moltisanti et al. 2019)	47.18	38.23	22.24	85.66	66.20	40.87	40.94	30.33	17.56	27.10	19.31	10.86
Test	Fixed segment	43.93	38.01	20.38	82.54	65.85	39.25	40.70	34.79	18.17	21.26	13.57	07.18
	(Moltisanti et al. 2019)	46.59	37.33	21.79	82.97	65.78	40.83	42.80	32.29	18.37	21.81	14.28	08.23

Table 6 Temporal action detection results in mAP (%)

Split	Baseline	Task	Mean Average Precision (mAP)					
			@0.1	@0.2	@0.3	@0.4	@0.5	Avg.
Val	BMN (Lin et al. 2019) + SlowFast (Feichtenhofer et al. 2019)	Verb	10.83	09.84	08.43	07.11	05.58	08.36
		Noun	10.31	08.33	06.17	04.47	03.35	06.53
		Act.	06.95	06.10	05.22	04.36	03.43	05.21
Test	BMN (Lin et al. 2019) + SlowFast (Feichtenhofer et al. 2019)	Verb	11.10	09.40	07.44	05.69	04.09	07.54
		Noun	11.99	08.49	06.04	04.10	02.80	06.68
		Act.	06.40	05.37	04.41	03.36	02.47	04.40

(in hours) than all datasets with higher average instances per video.

Evaluation Metrics In line with (Heilbron et al. 2015), we use mean Average Precision (mAP) by computing the average of the AP values for each class. A predicted segment matches a ground truth segment if their Intersection over Union (IoU) is greater than or equal to thresholds ranging from 0.1 to 0.5.

Baselines and Results We consider a two-stage baseline. Action proposals are first obtained using Boundary Matching Networks (BMN) (Lin et al. 2019), which are then classified using SlowFast (Feichtenhofer et al. 2019) (model trained as in Sect. 4.1). Results in Table 6 highlight that action detection is particularly challenging on this dataset, especially with respect to higher IoU thresholds. The qualitative example in Fig. 8 shows that our videos in EPIC-KITCHENS-100 contain actions of varying lengths, which adds further challenges.

4.4 Action Anticipation

Definition We aim to predict $(\hat{v}, \hat{n}, \hat{a})$ as the verb/noun/action classes of the action, by observing a video segment of arbitrary duration τ_o seconds (observation time) ending τ_a seconds (anticipation time) before the action's start, t_s. We set $\tau_a = 1$. We expect models addressing this task to reason on observed sequences of actions, the current state of the world (e.g., what objects are visible) and the possible goal of the camera wearer.

Related Datasets Table 4 also compares EPIC-KITCHENS-100 with other datasets used for action anticipation (Rohrbach et al. 2012; Patron-Perez et al. 2010; De Geest et al. 2016; Jiang et al. 2014; Kuehne et al. 2014; Stein and McKenna 2013; Li et al. 2015). Our dataset is the largest in hours and classes, and is unscripted, which is critical for meaningful anticipation models, and for in the wild testing.

Evaluation Metrics We report results using class-mean top-5 recall (Furnari et al. 2018). The top-k criterion accounts for uncertainty in future predictions, as with previous anticipation efforts (Koppula and Saxena 2016; Lee et al. 2017; Bhattacharyya et al. 2019). Class-mean allows for balancing the long-tail distribution.

Baselines and Results We use our prior work RU-LSTM (Furnari and Farinella 2020) as a baseline. In Table 7, RU-LSTM performs better for nouns compared to verbs, but shows that tail classes are particularly challenging for anticipation. Figure 9 demonstrates the baseline struggles where the next active noun/verb are ambiguous.

4.5 Unsupervised Domain Adaptation for Action Recognition

Definition Unsupervised Domain Adaptation (UDA) utilises a labelled source domain and learns to adapt to an unlabelled target domain. We use videos recorded in 2018 as the labelled source, and use newly collected videos as unlabelled target (i.e. without any of the accompanying annotations). The action recognition task itself follows the definition in
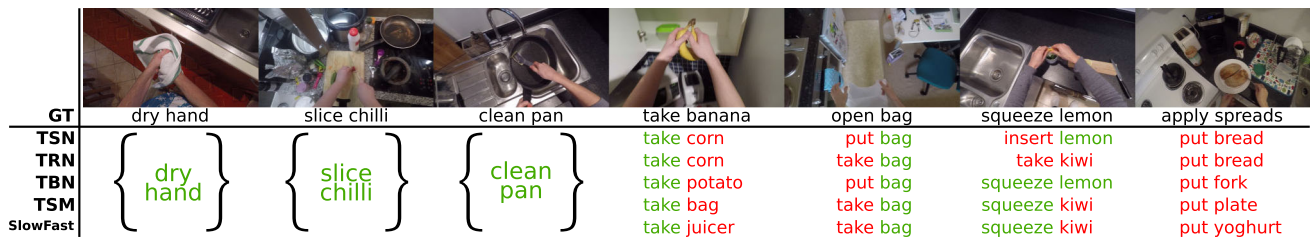
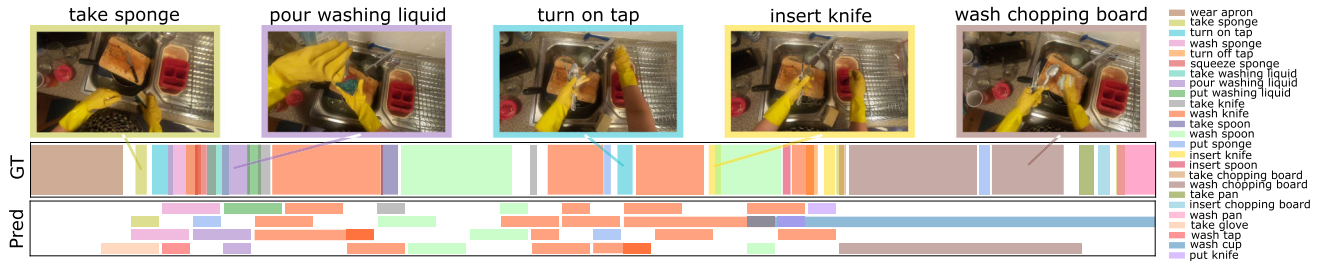**Fig. 7** Qualitative action recognition results for various baselines

Fig. 8 Qualitative results of action detection. Predictions with confidence > 0.5 are shown with colour-coded class labels (see legend). Since the baseline predicts overlapping segments, the predictions are displayed over four rows for ease of viewing

Table 7 Action anticipation results reported in class-mean top-5 recall (%)

| Split	Baseline	Overall			Unseen Participants			Tail Classes		
		Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Val	Chance	06.39	02.00	00.20	14.35	02.88	00.51	01.64	00.24	00.05
	RU-LSTM (Furnari and Farinella 2020)	27.76	30.76	14.04	28.78	27.22	14.15	19.77	22.02	11.14
Test	Chance	06.17	02.28	00.14	08.14	03.28	00.31	01.87	00.66	00.03
	RU-LSTM (Furnari and Farinella 2020)	25.25	26.69	11.19	19.36	26.87	09.65	17.56	15.97	07.92
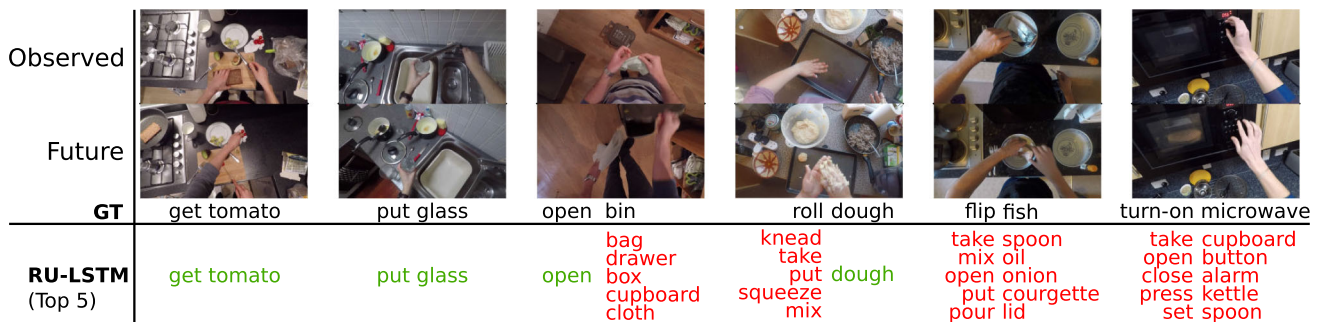
Fig. 9 Qualitative action anticipation results

Sect. 4.1. The difficulty of this challenge stems from the fact that the source and target domains come from distinct training distributions due to the collection of videos two years later . Changes in location, hardware and long-term temporal offsets are the main sources of the domain shift (see Sect. 3). A method which is able to perform this task well provides a number of practical benefits, most notably the elimination of labelling time and expense when collecting new videos, in the future.

Related Datasets UDA datasets have traditionally used images (Saenko et al. 2010; Venkateswara et al. 2017; Peng et al. 2017, 2019), with recent attempts to use video (Jamal et al. 2018; Chen et al. 2019; Qi et al. 2018) adapting across public datasets (e.g. UCF to Olympics). EPIC-KITCHENS-100 is the first to propose a within-dataset domain adaptation challenge in video. Video-based UDA raises additional challenges, such as aligning temporal information across domains (Jamal et al. 2018), attending to relevant transferable frames

Table 8 Comparison of domain adaptation classification datasets.

Dataset	Train	Test		Classes	SD	M	AAL	Year	Real/Syn
Image									
Office (Saenko et al. 2010)	4110	N/A		31	1	3	N/A	2010	Real
ImageCLEF (Caputo et al. 2014)	2400	600		12	5	1	N/A	2014	Real
Office ↔ Home (Venkateswara et al. 2017)	15500	N/A		65	4	1	N/A	2017	Real
VisDA-C (Peng et al. 2017)	280157	N/A		12	3	1	N/A	2017	Real/Syn
DomainNet (Peng et al. 2019)	363534	37706		345	6	1	N/A	2019	Real/Syn
	Source	Target	Test*						
Video									
UCF ↔ HMDB (small) (Xu et al. 2016)	482	350	150	5	2	1	4.7 ± 2.5	2018	Real
UCF ↔ Olympic (Jamal et al. 2018)	601	250	54	6	2	1	6.6 ± 4.5	2018	Real
UCF ↔ HMDB (full) (Chen et al. 2019)	1438	840	360	12	2	1	4.0 ± 5.8	2019	Real
IEMOCAP → AFEW (Qi et al. 2018)	6611	795	N/A	4	2	2	N/A	2018	Real
Kinetics ↔ Gameplay (Chen et al. 2019)	43378	2625	749	30	2	1	N/A	2019	Real/Syn
EPIC-KITCHENS-100	16115	26115	5909	3369	16	3	2.8 ± 5.2	2020	Real

Note that Test refers to 'Target Test' SD: Subdomains. M: Modalities. AAL: Average Action Length

(Chen et al. 2019), and avoiding non-informative background frames (Pan et al. 2020).

Table 8 shows EPIC-KITCHENS-100 provides several advantages over other video-based datasets: largest number of instances, classes, subdomains, and is multi-modal (Munro and Damen 2020). Additionally, it has a compound domain gaps resulting from the test of time (i.e. recording data two years later).

Splits This challenge assesses models' ability to adapt to additional footage without labels. We thus define the following splits; *Source*: labelled training data from 16 participants (collected in 2018) and *Target:* unlabelled footage from the same 16 participants collected in 2020. This ensures the gap in the domains is related to the capturing of the data 'two years later'. We further split target videos into: *Target Train* and *Target Test*. The first are unlabelled videos used during domain adaptation, while the second are videos used for evaluation, as in Peng et al. (2017). Number of action instances per split are reported in Table 8.

Evaluation We use the same evaluation metrics as Sect. 4.1 on Target Test.

Baselines and Results We present lower and upper bounds: "Source-Only", where labelled source data is used for training and no adaptation to target data is attempted, and two upper bounds: "Target-Only", where *labelled* target data is used and "Source+Target" where all training data is used with associated labels. Neither of these are UDA methods, but offer an insight into the domain gap.

Table 9 reports the results for the baselines. These use extracted features from TBN (Kazakos et al. 2019) trained on source. We use the code of Temporal Attentive Alignment (TA3N) (Chen et al. 2019), modified to consider multi-modal features (RGB, Flow and Audio), to report results. These show significant performance improvement when using multi-modal data compared to single modality models of RGB, Flow and Audio. The domain gap is evident when comparing the lower and upper bounds. TA3N is able to partially decrease this gap, providing a 2.5% improvement in verb accuracy and 2.4% in nouns when using multiple modalities. Recent work (Planamente et al. 2021) showed that RGB and Audio exhibit different levels of robustness to the domain gap in EPIC-KITCHENS-100. The best performing submissions for this challenge in 2021 exploited multi-modalities for domain adaptation (Yang et al. 2021; Plizzari et al. 2021). Fig. 10 visualises the multi-modal feature space showing limited overlap between source and target. TA3N aligns the features demonstrating the capability of UDA.

4.6 Multi-Instance Action Retrieval

Definition Given a query action segment, the aim of video-to-text retrieval is to rank captions in a gallery set, C, such that those with a higher rank are more *semantically relevant* to the action in the video. Conversely, text-to-video retrieval uses a query caption $c_i \in C$ to rank videos. Different from other challenges in Sect. 4, we here use the English-translated free-form captions from the narrations (Fig. 2b).

Table 9 Unsupervised domain adaptation results with lower (source-only) and the upper bounds of target-only and source+target

Modality	Baseline	Top-1 Acc. (%)			Top-5 Acc. (%)		
		Verb	Noun	Act.	Verb	Noun	Act.
RGB	Source-Only	32.8	21.2	10.7	72.6	43.9	22.6
	TA3N (Chen et al. 2019)	32.1	21.6	11.1	71.7	44.1	22.5
	Target-Only	*39.7*	*32.3*	*18.3*	*80.8*	*56.2*	*34.0*
	Source+Target	*41.1*	*33.0*	*18.8*	*80.4*	*58.5*	*35.2*
Flow	Source-Only	42.8	19.2	12.7	74.5	38.5	23.8
	TA3N (Chen et al. 2019)	43.2	20.1	12.8	74.5	41.2	25.0
	Target-Only	*53.8*	*26.7*	*20.2*	*84.2*	*49.1*	*34.5*
	Source+Target	*52.4*	*27.3*	*19.8*	*82.4*	*50.3*	*35.4*
Audio	Source-Only	31.4	12.8	8.5	64.8	28.4	16.0
	TA3N (Chen et al. 2019)	32.0	13.3	8.9	66.0	29.1	16.5
	Target-Only	*41.7*	*19.1*	*13.4*	*77.2*	*39.6*	*23.6*
	Source+Target	*41.8*	*19.8*	*13.8*	*77.1*	*40.6*	*24.3*
RGB+Flow +Audio	Source-Only	44.4	25.3	16.8	69.7	48.4	29.1
	TA3N* (Chen et al. 2019)	46.9	27.7	19.0	72.7	50.7	30.5
	Target-Only	*59.1*	*40.3*	*30.4*	*85.0*	*65.0*	*47.8*
	Source+Target	*59.4*	*41.9*	*31.3*	*85.3*	*66.6*	*49.2*

*Modified to consider multi-modal features
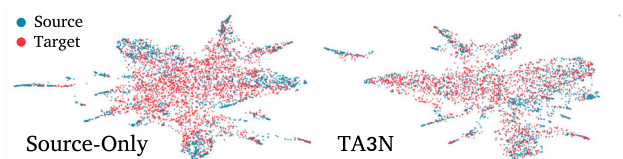
Fig. 10 UMAP (McInnes et al. 2018) of feature spaces shows better alignment through UDA baseline.

Splits We use the Train split from Table 1. As access to the captions are required for both video-to-text and text-to-video retrieval, the Val set is used for evaluating this challenge to allow the held-out Test set for all other challenges to remain intact. We consider all the videos in Val, and all unique captions, removing repeats.

Related Datasets In datasets that are commonly used for retrieval (Xu et al. 2016; Rohrbach et al. 2015; Zhou et al. 2017; Chen and Dolan 2011), captions are considered relevant if they were collected for the same video, and irrelevant otherwise. This common approach ignores the semantic overlap between captions of different videos that contain identical or similar actions. These datasets thus assume videos to be distinct from one another. In instructional video datasets (Zhou et al. 2017; Miech et al. 2019), the corresponding YouTube subtitle is only considered relevant, again ignor-

ing semantic overlap or similarities to other actions. Note that the large-scale HowTo100M [117] dataset has only been used for pre-training, due to being webly supervised and thus noisy. The dataset does not include a val/test set.

In this challenge, we use the class knowledge from Sect. 3 to define caption relevancy. This allows us to consider captions "put glass" and "place cup" as semantically relevant—an opportunity not available in other retrieval datasets.

Evaluation Metrics To evaluate this challenge, relevancy of a retrieved caption (or video) to the query item needs to be assessed. We consider the case where a query video contains the action of *someone cutting a pizza using a cutter*. We want captions: **a** "cutting a pizza using a cutter", **b** "cutting a pizza slice", **c** "slicing a pizza" to all be *more* relevant than **d** "cutting a pizza using a knife" which in turn is *more* relevant than both **e** "cutting a vegetable" or **f** "picking up a pizza slice". Critically, **g** "opening a fridge" should be considered irrelevant.

Mean Average Precision (mAP) has been used in other retrieval works (Wray et al. 2019; Rasiwasia et al. 2014; Kang et al. 2015; Cao et al. 2017), yet it only considers relevance between items to be binary. Because of this, (a–c) would be considered (equally) relevant captions. However, we would also like to consider non-binary relevance where

Table 10 Multi-Instance retrieval datasets

Dataset	Segments	Captions	Cap. Length	Cap./Seg.	Action Length	Rel. Caps.	Video Source	Cap. Source
MSR-VTT (Xu et al. 2016)	10,000	200,000	9.3	20.0	15.0s	20.0	YouTube	Collected
MSVD (Chen and Dolan 2011)	2,089	122,665	7.1	40.9	9.8s	40.9	YouTube	Collected
LSMDC (Rohrbach et al. 2015)	69,000	68,000	9.6	1.0	3.9s	1.0	Film	Script/AD
YouCook2 (Zhou et al. 2017)	13,829	13,829	8.8	1.0	19.6s	1.0	YouTube	Subtitle
EPIC-KITCHENS-100	76,885	19,803	3.0	1.0	3.1s	22.4	Unscripted	Narration

(d) is more relevant than (e) which in turn is more relevant than (g). We thus also report results using normalised Discounted Cumulative Gain (nDCG) (Järvelin and Kekäläinen 2002). This metric allows for non-binary relevance between captions. We define the relevance, $\mathcal{R}$, as the mean IoU of the verb and noun classes, giving a value between 0 and 1, where 0 is irrelevant (no overlap in verb/noun classes) and 1 is extremely relevant. From the example above, $1 = \mathcal{R}(a,a) \geq \mathcal{R}(a,b) = \mathcal{R}(a,c) \geq \mathcal{R}(a,d) \geq \mathcal{R}(a,e) = \mathcal{R}(a,f) \geq \mathcal{R}(a,g) = 0$. We then use $\mathcal{R}$ to calculate nDCG as in (Järvelin and Kekäläinen 2002) (see appendix C.6 for full definition).

Baselines and Results As in Sect. 4.5, we use TBN (Kazakos et al. 2019) features trained on the Train split. Table 11 provides results for two baselines and the chance lower bound. Multi-Layer Perceptron (MLP) uses a 2-layer perceptron to project both modalities into a shared action space with a triplet loss. Our previous work JPoSE (Wray et al. 2019) disentangles captions into verb, noun and action spaces learned with a triplet loss. JPoSE sees a significant boost in performance over MLP. Figure 11 shows qualitative retrieval results on four examples using both MLP and JPoSE for text-to-video retrieval. JPoSE is able to retrieve more correct videos than MLP, but both methods still struggle on longer captions. Importantly, this dataset offers the first opportunity for action retrieval that considers semantic similarity.

5 Conclusion and Future Work

We presented our large-scale egocentric dataset EPIC-KITCHENS-100, through an annotation pipeline that is scalable and is of higher quality than previous approaches. We defined six challenges, providing leaderboard baselines. Dataset and leaderboards are available at http://epic-kitchens.github.io.

These 6 challenges have been chosen to facilitate progress in open topics within video understanding. They also highlight interesting parts of our collection and annotation pipeline. For example, retrieval uses our free-form captions, while unsupervised domain adaptation for action recognition builds on collecting footage two years later. Our dense annotations of overlapping actions make detection in long untrimmed videos particularly challenging. While this paper addresses each challenge independently, successful methods that address one challenge (e.g. detection) are likely to prove advantageous for better performance in another (e.g. anticipation). Combining all challenges with unsupervised domain adaptation would enable future deployment in new environments without additional labels.

In publishing this manuscript we hope that people can not only utilise this large-scale dataset in their ongoing research, but also build on our novel pipeline in collecting our dataset.

Table 11 Multi-Instance retrieval results

Baseline	mAP			nDCG		
	vid→txt	txt→vid	Avg.	vid→txt	txt→vid	Avg.
Chance	5.7	5.6	5.7	10.8	10.9	10.9
MLP	43.0	34.0	38.5	50.1	46.9	48.5
JPoSE (Wray et al. 2019)	49.9	38.1	44.0	55.5	51.6	53.5

Fig. 11 Qualitative results for text-to-video action retrieval. Top 3 retrieved videos and the semantic relevancy $\mathcal{R}$ of the top 50 retrievals (red: irrelevant, green: relevant) (Color figure online)

The proposed 'pause-and-talk' narrator, publicly available, as well as our visually-supported transcription interfaces can prove advantageous for other large-scale collection efforts.

Data Release Statement Dataset sequences, extracted frames and optical flow are available under Non-Commercial Government Licence for public sector information at the University of Bristol data repository: http://dx.doi.org/10.5523/bris.2g1n6qdydwa9u22shpxqzp0t8m
Annotations, models, evaluation scripts, challenge leaderboards and updates are available at: http://epic-kitchens.github.io

Appendices

A Video Demonstration

We provide a video demonstration of our annotation pipeline and six challenges. Our video utilises a single sequence, showcasing the annotation pipeline first, as the sequence progresses. We demonstrate the 'pause-and-talk' narrator, transcription and translation steps, then parsing and class mapping. We then showcase the two automatic annotations provided with our dataset.

The video demonstrates predictions from our six challenges. This showcases baseline results, but on a training sequence demonstrating 'near perfect' performance as opposed to current baseline performance. This aims to highlight the potential of EPIC-KITCHENS-100 and the link between these challenges. Our Video demonstration is available at: https://youtu.be/MUlyXDDzbZU

B Further Collection Details

In this section we provide further details of how EPIC-KITCHENS-100 was collected including comparing to the

annotation pipeline from our previous work (Damen et al. 2018).

Camera Settings for Collection Head mounted GoPro Hero 7 was used for data collection filming at 50fps with video stabilisation. Our choice of 50 fps avoids overhead light flickering visible in Damen et al. (2018) that occurs due to the difference between frame rates and the national grid frequency.

Narration 'pause-and-talk' interface Fig. 12 contains a more detailed look at our proposed 'pause-and-talk' narrator. Annotators had a number of options to help with the recording, including whether or not to hear the audio from the captured video while narrating, and the ability to change the speed of the video. They could also play, redo or delete recordings they had already made.

As mentioned in Sect. 2, this led to denser and more correct annotations, as annotators were able to pause the video while providing annotations, avoiding any missed annotations of critical actions.

Transcription Thanks to our 'pause-and-talk' narrator, each audio clip contained a single action narration, whereas formerly speech chunks were combined into 30 second clips. In Damen et al. (2018), Amazon Mechanical Turk (AMT) workers had to translate and transcribe this audio narration in a single step. To ensure correctness and consistency, we split the transcription from the translation steps. The set of non-English transcriptions was first agreed by multiple annotators and then translated in one go by a hired translator.

Additionally, we provided images during the transcription step centred around the timestamp collected by the 'pause-and-talk' Narrator at $\{-0.25s, 0s, +0.25s\}$ to improve context (see Fig. 1b).

Temporal Annotator Previously, initial start/end times were obtained by automatic alignment of captions using YouTube automatic subtitling API. This is problematic as it assumes action length is the same as the narration length. We adopt a different approach here starting from our accurate single timestamps produced by our proposed 'pause-and-talk' narrator. We developed a temporal segment annotation interface (see Fig 1d), where annotators start from this rough-time stamp and annotate the start/end time. We also increased the number of annotators per segment to 5, compared to 4 used in Damen et al. (2018). This resulted in higher agreements between annotators.

C Challenges' Implementation Details

In this section we include the implementation and training details for all of the baselines, to enable replication of our results. Additionally, for some challenges, further details are provided such as definition of evaluation metrics.

C.1 Action Recognition

Implementation and Training Details We use our publicly available PyTorch (Paszke et al. 2019) model definitions of TSN (Wang et al. 2016), TRN (Zhou et al. 2018) and TSM (Lin et al. 2019). We use ResNet-50 backbones for all models with publicly available initialisations - these are ImageNet weights for TSN and TRN and Kinetics weights for TSM. We train two instances of each model: one with 8 RGB frames as input, and the other with 8 stacks of 5 (u, v) flow fields computed using TV-L_1 (Zach et al. 2007). We use two-way output in the last layer, one to predict verbs and the other to predict nouns with an average verb/noun loss. Actions are predicted as the most likely verb-noun combinations computed by combining softmaxed verb/noun scores.

We train each model for 80 epochs using SGD with momentum 0.9 and a learning rate of 0.01 decayed at epochs 20 and 40 by a factor of 10. TSN and TRN models are trained on 8 GPUs with a batch-size of 128, whereas TSM used a batch-size of 64 on 4 GPUs. We apply a weight decay of 0.0005 to all weights in the models, drop out with $p = 0.7$, and clipping gradients above 20. We use center-crop evaluation. The RGB and optical flow models are trained individually, and predictions are averaged pre-softmax during inference.

For TBN, we use the publicly available PyTorch (Paszke et al. 2019) model from Kazakos et al. (2019). We train using a batch size of 64, 6 segments, and drop the learning rate at epoch 40 and 60. All unspecified hyperparameters remain unchanged.

For SlowFast (Feichtenhofer et al. 2019), we use the publicly available PyTorch (Paszke et al. 2019) model. We modify the model to have a two-way output for verbs and nouns, and train it with the average verb-noun loss. We use the SlowFast 8x8, ResNet-50 backbone, initialised from Kinetics pretrained weights also provided by Feichtenhofer et al. (2019). A 1 second clip randomly sampled from the video is used as input to the model during training. We train for 30 epochs using SGD with momentum 0.9 and a learning rate of 0.01 decayed at epochs 20 and 25 by a factor of 10. The model is trained on 8 GPUs with a batch-size of 32, using a weight decay of 0.0001 to all weights in the model and drop out with $p = 0.5$. We freeze all batch-normalisation layers' parameters and statistics during training. During testing, we uniformly sample 10 clips (1s each) from each video, and a single center crop per clip, and average their predictions.
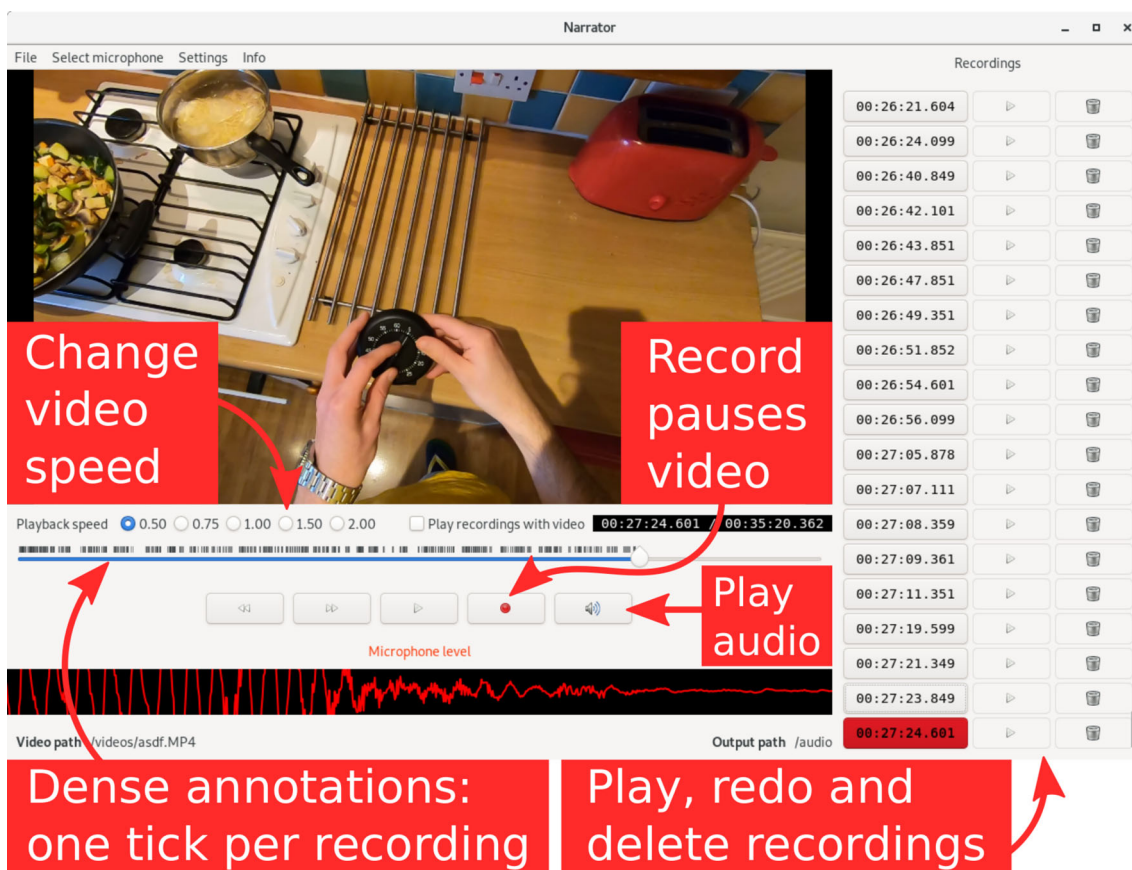
Fig. 12 Components of our 'pause-and-talk' annotation tool

C.2 Weakly-Supervised Action Recognition

Implementation and Training Details We use our publicly available PyTorch (Paszke et al. 2019) code from Moltisanti et al. (2019) for both baselines. This uses TSN (Wang et al. 2016) with Inception backbone and batch normalisation (Ioffe and Szegedy 2015), pre-trained on Kinetics-400 (Carreira and Zisserman 2017). Predictions employ standard late-fused two-stream approach at test time (RGB and Flow models are trained independently). This uses 25 RGB frames (or optical flow stacks) for testing.

We set a length of 5 seconds for the fixed-length segment baseline. For this baseline, frames are sampled randomly from equally sized segments (as proposed in Wang et al. (2016)). For the baseline from Moltisanti et al. (2019) training frames are selected using the sampling distributions which are iteratively updated. For both baselines we sample 5 frames for training. The ADAM (Kingma and Ba 2014) optimiser is used with initial learning rate equal to 0.0001 halved twice during training, and report results after 80 epochs. We changed the parameters from Moltisanti et al. (2019) as follows: $w = 2.5$ seconds and $s = 0.75$, updating the distributions every 5 epochs with $(\lambda_c, \lambda_w, \lambda_s) = (0.5, 0.25, 0.25)$.

We set CL $h = 1$ and CL $z = 0.25$. Update proposals are generated with $\tau \in \{0.5, 0.85\}$, discarding proposals with length less than 10 frames.

C.3 Action Detection

Implementation and Training Details We train Boundary Matching Network (BMN) (Lin et al. 2019) using the publicly available implementation to produce temporal action proposals.[6] BMN is trained using TSN-based features, as in action recognition. As proposed in Lin et al. (2019), we rescale the feature sequence of each video to the length of the observation window l_ω. Since the proposed dataset contains videos of different lengths, we choose a large observation window $l_\omega = 400$ and set the maximum action length to $D = 400$. To limit the amount of memory required at training time, we set the number of sample points to $N = 4$. We train one model on the Train set for 9 epochs, which maximizes performance on Val. We use this model to report on both Val and Test. We apply Soft Non-Maximum Suppression with the parameters suggested in Lin et al. (2019) to

---

[6] https://github.com/JJBOY/BMN-Boundary-Matching-Network

reduce the number of overlapping proposals and retain the top scoring 1, 000 instances per video.

Each proposal is then classified using the SlowFast Network with implementation details as in Section C.1. Note that we classify proposals on the validation set using the SlowFast model trained only on the training set, whereas we classify proposals on the test set using the model trained on the union of the training and validation sets.

C.4 Action Anticipation

Implementation and Training Details We follow our prior work (Furnari and Farinella 2020) training a TSN model to extract RGB and Flow features, using the same hyperparameters recommended in Furnari and Farinella (2020). The RGB model has been trained for 95 epochs, while the optical flow branch has been trained for 132 epochs, which maximise performance on Val. Object-based features are extracted running the object detector from Furnari and Farinella (2020), trained on manually-annotated object bounding boxes from our previous edition Damen et al. (2018). The RU-LSTM model is trained using the provided implementation with SGD and a fixed learning rate of 0.01. The single-modality RGB, optical flow and object branches are pre-trained with sequence completion respectively for 88, 95, and 98 epochs, then fine-tuned for the anticipation task for 86, 81 and 7 epochs respectively. The full architecture with modality attention is trained for 29 epochs. These maximise performance on Val. All other parameters are kept as their default values in the public code from Furnari and Farinella (2020), The same model is used to report both on Val and Test.

Impact of current action on anticipation Predicting a future action given the currently observed one provides a strong prior. To assess this, we created three co-occurrence matrices for verbs, nouns and actions. Each matrix M is constructed such that $M[i, j]$ reports the number of times class j is observed after class i in the training set considering $\tau_a = 1$ as the anticipation time. At test time, we rely on the last observed action i to predict the most frequent 5 actions following i (corresponding to the 5 largest values of the ith row of M). Note that this calculation requires knowledge of the observed action from the ground truth, thus cannot be considered a baseline, as it cannot be replicated in inference. We found that this oracle knowledge of the current action obtains 20.84%, 25.00% and 8.92% for Top-5 verb, noun and action labels respectively on the validation set. These numbers are significantly larger than the chance baseline (6.39%, 2.00%, 0.20%) from Table 7 but still lower than the ones of the RU-LSTM baseline (27.76%, 30.76%, 14.04%). These results suggest that, while the prior is indeed a strong one, as you would hope for meaningful sequences of actions, the considered baseline is going beyond recognising the current action and applying an action sequence prior.

C.5 Unsupervised Domain Adaptation (UDA) for Action Recognition

Validation Splits for Hyper-parameter Tuning As the target domain is unlabelled, no labelled data is available for hyper-parameter tuning. Therefore, we split the training data to define a *Source Val* and *Target Val* splits with data collected by 4 of the 16 participants. Of these, 2 participants are of returning kitchens and 2 of changing kitchens. The *Source Train* and *Target Train* are thus composed of the 12 remaining participants.

For hyper-parameter tuning, models are trained on labelled data from *Source Val* and unlabelled from *Target Val*. The performance on *Target Val* can be used to asses the impact of different hyper-parameters.

To obtain the results for the leaderboard and accompanying challenge, a new model is trained on *Source Train* and unlabelled *Target Train*, using the hyper-parameters optimised from the validation split. This model is evaluated on *Target Test* to obtain results.

Note on zero-shot actions Due to the unscripted nature of the data collection, a negligible number of verb and noun classes in the target domain are not present in the source domain, 0.2% and 2.3% respectively. We have not removed these to maintain the same splits used in other challenges. Additionally, 9.46% actions (exact verb-noun combinations) did not exist in the targets domain, these are referred to as the zero-shot actions. Note it is still possible to predict these actions as both verbs and nouns were present in the source domain.

Implementation and Training Details We train the TBN feature extractor on the union of *Source Train* and *Source Val*. We make these features publicly available. We use the available code from Chen et al. (2019), to train and evaluate 'Source-Only' as well as 'TA3N' baselines. We modify the code to consider multi-modal input, by concatenating the features from all modalities as input. This automatically increased the number of parameters in the first fully connected layer.

We improve the performance of TA3N by initialising the domain discriminators before the gradients are reversed and back-propagated. In our implementation, the domain discriminators' hyper-parameters are annealed similar to that in Ganin et al. (2016):

$$\eta = \frac{2}{1 + exp(-p)} - 1 \qquad (1)$$

where p is the training progress that linearly increases from 0 to 1. The domain discriminator hyperparameters are annealed up to the value specified in TA3N, i.e. $\lambda^s = 0.75\eta$, $\lambda^r = 0.5\eta$ and $\lambda^t = 0.75\eta$. The weighting of the categorical entropy on the target domain is set to $\gamma = 0.003$. Models are trained for 30 epochs at a learning rate of $3e^{-3}$ reduced by a factor of 10 at epochs 10 and 20.

C.6 Multi-Instance Action Retrieval

Evaluation Metrics We define the Relevance $\mathcal{R}$ between a video, x_i, and a caption, c_j, as given by the averaged Intersection-over-Union of the verb and noun classes:

$$\mathcal{R}(x_i, c_j) = \frac{1}{2}\left(\frac{|x_i^v \cap c_j^v|}{|x_i^v \cup c_j^v|} + \frac{|x_i^N \cap c_j^N|}{|x_i^N \cup c_j^N|}\right) \quad (2)$$

where x_i^v is the set of verb classes in the video and c_j^N is the set of noun classes in the caption.

The nDCG can be calculated for a query video, x_i, and the ranked list of gallery captions, C_r, as the Discounted Cumulative Gain (DCG) over the Ideal Discounted Cumulative Gain (IDCG):

$$nDCG(x_i, C_r) = \frac{DCG(x_i, C_r)}{IDCG(x_i, C_r)} \quad (3)$$

with the DCG being given by:

$$DCG(x_i, C_r) = \sum_{j=1}^{|C_r|} \frac{\mathcal{R}(x_i, c_j)}{log(j+1)} \quad (4)$$

To calculate the $IDCG(x_i, C_r)$, we need the ground truth ranking between video x_i and captions C_r. To do this, we first find the relevance between video x_i and every caption in C_r as follows: $\{\mathcal{R}(x_i, c_j); \ \forall c_j \in C_r)\}$. We then construct $\hat{C}_r$, the ground truth ranking of captions, by sorting these in descending order of relevance. Note that if $\mathcal{R}(x_i, c_j) = \mathcal{R}(x_i, c_k)$ then c_j and c_k are ordered based on their unique ID due to the stable sort used, and similarly for the method to be evaluated. Finally, the $IDCG$ is calculated using $IDCG(x_i, C_r) = DCG(x_i, \hat{C}_r))$.

$nDCG$ can be similarly defined for a query caption, c_i and a gallery set of videos X_r.

Implementation and Training Details For video features we use 25 RGB, Flow and Audio features extracted uniformly from TBN (Kazakos et al. 2019). We make these features publicly available. Features from each modality are temporally averaged and then concatenated to provide the final feature vector for each video, with size 3072. Text features come from word2vec (Mikolov et al. 2013) trained on the wikipedia corpus with an embedding space of size 100.

The MLP baseline uses a 2 layer perceptron which projects both the visual and textual features into the same embedding space. We set the final embedding size to 512 and the size of the hidden units is 1280 and 78 for visual/textual respectively (halfway between initial feature size and output space size). MLP is trained for 100 epochs with a batch size of 64 and a learning rate of 0.01. Triplets are sampled randomly using the semantic relevance used when calculating mAP/nDCG (*i.e.* verb and noun class are identical), with triplets being sampled every 10 iterations. The triplet loss terms for all four pairs of modalities are set to 1.0, apart from the the text-to-visual weight which is assigned a weight of 2.0.

We use our public code of JPoSE (Wray et al. 2019) . Each Part-of-Speech embedding is modelled off of the MLP baseline, but using the part-of-speech relevancies defined in Wray et al. (2019) (e.g. for the verb embedding the verb class between two captions must be the same). The final embeddings are concatenated and fed into a final fully connected layer with shared weights for the action embedding. The verb and noun embedding spaces have an output embedding size of 256, with the resulting action embedding space having an output size of 512. Triplets are independently resampled (randomly) every 10 epochs. A batch size of 64 is used with a learning rate of 0.01 and the model is trained for 100 epochs.

References

Bearman A, Russakovsky O, Ferrari V, & Fei-Fei L (2016) What's the point: semantic segmentation with point supervision. In *ECCV*

Bhattacharyya A, Fritz M, & Schiele B (2019) Bayesian prediction of future street scenes using synthetic likelihoods. In *ICLR*

Bojanowski P, Lajugie R, Bach F, Laptev I, Ponce J, Schmid C, & Sivic J (2014) Weakly supervised action labeling in videos under ordering constraints. In *ECCV*

Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, & Beijbom O (2019) nuScenes: A multimodal dataset for autonomous driving. arXiv

Cao Y, Long M, Wang J, & Yu P (2017) Correlation hashing network for efficient cross-modal retrieval. In *BMVC*

Caputo B, Müller H, Martinez-Gomez J, Villegas M, Acar B, Patricia N, Marvasti N, Üsküdarlı S, Paredes R, Cazorla M, et al. (2014) Imageclef 2014: Overview and analysis of the results. In: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer 192–211

Carlevaris-Bianco, N., Ushani, A. K., & Eustice, R. M. (2016). University of Michigan North Campus long-term vision and lidar dataset. *Int J Robotics Res, 35*(9), 1023–1035.

Carreira J, & Zisserman A (2017) Quo Vadis, action recognition? A new model and the Kinetics dataset. In *CVPR*

Carreira J, Noland E, Hillier C, & Zisserman A (2019) A short note on the Kinetics-700 human action dataset. arXiv

Chang C, Huang DA, Sui Y, Fei-Fei L, & Niebles JC (2019) D3TW: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*

Chen D, & Dolan, W (2011) Collecting highly parallel data for paraphrase evaluation. In NAACL-*HLT*

Chen MH, Kira Z, AlRegib G, Yoo J, Chen R, & Zheng J (2019) Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*

Chéron G, Alayrac J, Laptev I, & Schmid C (2018) A flexible model for training action localization with varying levels of supervision. In *NeurIPS*

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, & Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In *CVPR*

Damen D, Doughty H, Farinella GM, Fidler S, Furnari A, Kazakos E, Moltisanti D, Munro J, Perrett T, Price W, & Wray M (2018) Scaling egocentric vision: The EPIC-KITCHENS dataset. In *ECCV*

Damen D, Leelasawassuk T, Haines O, Calway A, & Mayol-Cuevas W (2014). You-do, I-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*

De Geest R, Gavves E, Ghodrati A, Li Z, Snoek C, & Tuytelaars T (2016) Online action detection. In *ECCV*

De La Torre F, Hodgins J, Bargteil A, Martin X, Macey J, Collado A, & Beltran P (2008) Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) database. In *Robotics Institute*

Deng J, Dong W, Socher R, Li LJ, Li K, & Fei-Fei L (2009) Imagenet: A large-scale image database. In *CVPR*

Ding L, & Xu C (2018) Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*

Fathi A, Li Y, & Rehg J (2012) Learning to recognize daily actions using gaze. In *ECCV*

Feichtenhofer C, Fan H, Malik J, & He K (2019) SlowFast networks for video recognition. In *ICCV*

Furnari A, & Farinella GM (2020) Rolling-unrolling LSTMs for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*

Furnari A, Battiato S, & Farinella GM (2018) Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *ECCVW*

Ganin Y, Ustinova E, Ajakan, H, Germain P, Larochelle H, Laviolette F, Marchand M, & Lempitsky V (2016) Domain-adversarial training of neural networks. *JMLR*

Geiger A, Lenz P, & Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*

Gong B, Shi Y, Sha F, & Grauman K (2012) Geodesic Flow Kernel for Unsupervised Domain Adaptation. In *Computer Vision and Pattern Recognition*

Gorban A, Idrees H, Jiang YG, Zamir AR, Laptev I, Shah M, & Sukthankar R (2015). THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/

Goyal R, Kahou SE, Michalski V, Materzynska J, Westphal S, Kim H, Haenel V, Fründ I, Yianilos P, Mueller-Freitag M, Hoppe F, Thurau C, Bax I, Memisevic R (2017) The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*

Gu C, Sun C, Ross DA, Vondrick C, Pantofaru C, Li Y, Vijayanarasimhan S, Toderici G, Ricco S, Sukthankar R, Schmid C, & Malik J (2018) AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*

Gupta S, & Malik J (2016) Visual semantic role labeling. In *CVPR*

Gygli M, & Ferrari V (2019) Efficient object annotation via speaking and pointing. *IJCV*

He K, Girshick R, & Dollár P (2019) Rethinking ImageNet pre-training. In *ICCV*

He K, Gkioxari G, Dollár P, & Girshick R (2017) Mask R-CNN. In *ICCV*

Heckman, J. J. (1979). Sample Selection Bias as a Specification Error. *Econometrica, 47*(1), 153–161.

Heilbron FC, Escorcia V, Ghanem B, & Niebles JC (2015) ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*

Heilbron FC, Lee JY, Jin H, & Ghanem B (2018) What do i annotate next?, An empirical study of active learning for action localization. In *ECCV*

Honnibal M, & Montani I (2017) spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing

Hsu HK, Yao CH, Tsai YH, Hung WC, Tseng HY, Singh M, & Yang MH (2020) Progressive domain adaptation for object detection. In *Winter Conference on Applications of Computer Vision*

Huang X, Cheng X, Geng Q, Cao B, Zhou D, Wang P, Lin Y, & Yang R (2018) The apolloscape dataset for autonomous driving. In *CVPRW*

Huang DA, Fei-Fei L, & Niebles JC (2016) Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*

Ioffe S, & Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*

Jamal A, Namboodiri VP, Deodhare D, & Venkatesh K (2018) Deep domain adaptation in action space. In *BMVC*

Järvelin K, & Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. *TOIS*

Jiang YG, Liu J, Zamir AR, Toderici G, Laptev I, Shah M, & Sukthankar R (2014) THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/

Kang C, Xiang S, Liao S, Xu C, & Pan C (2015) Learning consistent feature representation for cross-modal multimedia retrieval. *TMM*

Karpathy A, & Fei-Fei L (2015) Deep Visual-Semantic Alignments for Generating Image Descriptions. In *CVPR*

Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, Suleyman M, & Zisserman A (2017) The Kinetics human action video dataset. arXiv

Kazakos E, Nagrani A, Zisserman A, & Damen D (2019) EPIC-Fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*

Kingma DP, & Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Koppula HS, & Saxena A (2016) Anticipating human activities using object affordances for reactive robotic response. *TPAMI*

Krishna R, Hata K, Ren F, Fei-Fei L, & Niebles JC (2017) Dense-captioning events in videos. In *ICCV*

Kuehne H, Arslan A, & Serre T (2014) The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*

Kuehne H, Jhuang H, Garrote E, Poggio T, & Serre T (2011) HMDB: a large video database for human motion recognition. In *ICCV*

Lea C, Flynn MD, Vidal R, Reiter A, & Hager GM (2017) Temporal convolutional networks for action segmentation and detection. In *CVPR*

Lee N, Choi W, Vernaza P, Choy C, Torr PHS, & Chandraker M (2017) DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*

Li J, Lei P, & Todorovic S (2019) Weakly supervised energy-based learning for action segmentation. In *ICCV*

Li Y, Ye Z, & Rehg JM (2015) Delving into egocentric actions. In *CVPR*

Lin J, Gan C, & Han S (2019) TSM: Temporal shift module for efficient video understanding. In *ICCV*

Lin T, Liu X, Li X, Ding E, & Wen S (2019) BMN: Boundary-matching network for temporal action proposal generation. In *ICCV*

Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, & Zitnick CL (2014) Microsoft COCO: Common objects in context. In *ECCV*

Liu D, Jiang T, & Wang Y (2019) Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*

Liu Z, Miao Z, Zhan X, Lin D, Yu SX, & Icsi, UCB (2020) Open Compound Domain Adaptation. In *Computer Vision and Pattern Recognition*

Maddern, W., Pascoe, G., Linegar, C., & Newman, P. (2017). 1 year, 1000 km: the Oxford RobotCar dataset. *Int J Robot Res, 36*(1), 3–15.

Mahdisoltani F, Berger G, Gharbieh W, Fleet D, & Memisevic R (2018) On the effectiveness of task granularity for transfer learning. arXiv

Marszalek M, Laptev I, & Schmid C (2009) Actions in context. In *CVPR*

McInnes L, Healy J, & Melville J (2018) UMAP: Uniform manifold approximation and projection for dimension reduction. arXiv

Mettes P, Koelma DC, & Snoek CGM (2016) The ImageNet shuffle: Reorganized pre-training for video event detection. In *ICMR*

Mettes P, Van Gemert JC, & Snoek CG (2016) Spot on: Action localization from pointly-supervised proposals. In *ECCV*

Miech A, Zhukov D, Alayrac JB, Tapaswi M, Laptev I, & Sivic J (2019) HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*

Mikolov T, Chen K, Corrado G, & Dean J (2013) Efficient estimation of word representations in vector space. In *ICLR*

Moltisanti D, Fidler S, & Damen D (2019). Action recognition from single timestamp supervision in untrimmed videos. In *CVPR*

Moltisanti D, Wray M, Mayol-Cuevas W, & Damen D (2017) Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *ICCV*

Monfort M, Vondrick C, Oliva A, Andonian A, Zhou B, Ramakrishnan K, Bargal SA, Yan T, Brown L, Fan Q, & Gutfreund D (2020) Moments in Time dataset: One million videos for event understanding. *TPAMI*

Munro J, & Damen D (2020) Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*

Narayan S, Cholakkal H, Khan F, & Shao L (2019) 3C-Net: Category count and center loss for weakly-supervised action localization. In *ICCV*

Neuhold G, Ollmann T, Bulo SR, & Kontschieder P (2017) The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*

Nguyen P, Liu T, Prasad G, & Han B (2018). Weakly supervised action localization by sparse temporal pooling network. In *CVPR*

Nguyen P, Ramanan D, & Fowlkes C (2019) Weakly-supervised action localization with background modeling. In *ICCV*

Noroozi M, & Favaro P (2016) Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*

Oberdiek P, Rottmann M, & Fink GA (2020) Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation. In *Computer Vision and Pattern Recognition Workshops*

Pan B, Cao Z, Adeli E, & Niebles JC (2020) Adversarial cross-domain action recognition with co-attention. In *AAAI*

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, & Chintala S (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach H, Larochelle H, Beygelzimer A, dÁlché-Buc F, Fox E, & Garnett R, eds. Advances in Neural Information Processing Systems 32. Curran Associates, Inc. 8024–8035

Patron-Perez A, Marszalek M, Zisserman A, & Reid I (2010) High Five: Recognising human interactions in TV shows. In *BMVC*

Peng X, Bai Q, Xia X, Huang Z, Saenko K, & Wang B (2019) Moment matching for multi-source domain adaptation. In *ICCV*

Peng X, Usman B, Kaushik N, Hoffman J, Wang D, & Saenko K (2017) VisDA: The visual domain adaptation challenge. arXiv

Perazzi F, Pont-Tuset J, McWilliams B, Gool LV, Gross M, & Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*

Pirsiavash H, & Ramanan D (2012) Detecting activities of daily living in first-person camera views. In *CVPR*

Planamente M, Plizzari C, Alberti E, & Caputo B (2021) Cross-domain first person audio-visual action recognition through relative norm alignment. arXiv preprint arXiv:2106.01689

Plizzari C, Planamente M, Alberti E, & Caputo B (2021). Polito-iit submission to the epic-kitchens-100 unsupervised domain adaptation challenge for action recognition. arXiv preprint arXiv:2107.00337

Qi F, Yang X, & Xu C (2018) A unified framework for multimodal domain adaptation. In *ACM-MM*

Rasiwasia N, Mahajan D, Mahadevan V, & Aggarwal G (2014) Cluster canonical correlation analysis. In *AISTATS*

Richard A, Kuehne H, Iqbal A, & Gall J (2018) NeuralNetwork-Viterbi: A framework for weakly supervised video learning. In *CVPR*

Rohrbach, M., Amin, S., Andriluka, M., & Schiele, B. (2012). A database for fine grained activity detection of cooking activities. In *CVPR*

Rohrbach A, Rohrbach M, Tandon N, & Schiele B (2015) A dataset for movie description. In *CVPR*

Roth J, Chaudhuri S, Klejch O, Marvin R, Gallagher A, Kaver L, Ramaswamy S, Stopczynski A, Schmid C, Xi Z, et al. (2019). AVA-ActiveSpeaker: An audio-visual dataset for active speaker detection. arXiv

Saenko K, Kulis B, Fritz M, & Darrell T (2010) Adapting visual category models to new domains. In *ECCV*

Saenko K, Kulis B, Fritz M, & Darrell T (2010) Adapting Visual Category Models to New Domains. In *European Conference on Computer Vision*

Shan D, Geng J, Shu M, & Fouhey DF (2020) Understanding human hands in contact at internet scale. In *CVPR*

Sigurdsson GA, Gupta A, Schmid C, Farhadi A, & Alahari K (2018) Charades-ego: A large-scale dataset of paired third and first person videos. In *ArXiv*

Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, & Gupta A (2016) Hollywood in Homes: Crowdsourcing data collection for activity understanding. In *ECCV*

Silberman N, Hoiem D, Kohli P, & Fergus R (2012) Indoor segmentation and support inference from RGBD images. In *ECCV*

Singh KK, & Lee YJ (2017). Hide-and-Seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*

Soomro K, Zamir AR, & Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv

Stein S, & McKenna SJ (2013). Combining Embedded Accelerometers with Computer Vision for Recognizing Food Preparation Activities. In *International Joint Conference on Pervasive and Ubiquitous Computing*

Stein S, & McKenna S (2013) Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*

Torralba A, & Efros AA (2011) Unbiased look at dataset bias. In *CVPR* 2011

Ueberla JP (1997) Domain adaptation with clustered language models. In *International Conference on Acoustics, Speech and Signal Processing*

Venkateswara H, Eusebio J, Chakraborty S, & Panchanathan S (2017) Deep hashing network for unsupervised domain adaptation. In *CVPR*

Vondrick C, Shrivastava A, Fathi A, Guadarrama S, & Murphy K (2018) Tracking emerges by colorizing videos. In *ECCV*

Wang L, Xiong Y, Lin D, & Van Gool L (2017) Untrimmednets for weakly supervised action recognition and detection. In *CVPR*

Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, & Gool LV (2016) Temporal Segment Networks: Towards good practices for deep action recognition. In *ECCV*

Weinzaepfel P, Martin X, & Schmid C (2016) Human action localization with sparse spatial supervision. arXiv

Wray M, Larlus D, Csurka G, & Damen D (2019) Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*

Wulfmeier M, Bewley A, & Posner I (2018) Incremental Adversarial Domain Adaptation for Continually Changing Environments. In *International Conference on Robotics and Automation*. 4489–4495

Xu J, Mei T, Yao T, & Rui Y (2016) MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*

Xu T, Zhu F, Wong EK, & Fang Y (2016) Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition. *IMAVIS*

Yang L, Huang Y, Sugano Y, & Sato Y (2021) Epic-kitchens-100 unsupervised domain adaptation challenge for action recognition 2021: Team m3em technical report. arXiv preprint arXiv:2106.10026

Yeung S, Russakovsky O, Jin N, Andriluka M, Mori G, & Fei-Fei L (2017) Every Moment Counts: Dense detailed labeling of actions in complex videos. *IJCV*

Yogamani S, Hughes C, Horgan J, Sistu G, Varley P, O'Dea D, Uricár M, Milz S, Simon M, Amende K et al. (2019) Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *ICCV*

Yu F, Xian W, Chen Y, Liu F, Liao M, Madhavan V, & Darrell T (2018) BDD100K: A diverse driving video database with scalable annotation tooling. arXiv

Zach C, Pock T, & Bischof H (2007) A duality based approach for realtime TV-L1 optical flow. In *Pattern Recognition*

Zamir AR, Sax A, Shen W, Guibas L, Malik J, & Savarese S (2018) Taskonomy: Disentangling task transfer learning. In *CVPR*

Zhai M, Bessinger Z, Workman S, & Jacobs N (2017) Predicting Ground-Level Scene Layout from Aerial Imagery. In *Computer Vision and Pattern Recognition*

Zhai X, Puigcerver J, Kolesnikov A, Ruyssen P, Riquelme C, Lucic M, Djolonga J, Pinto AS, Neumann M, Dosovitskiy A, Beyer L, Bachem O, Tschannen M, Michalski M, Bousquet O, Gelly S, & Houlsby N (2019) A large-scale study of representation learning with the visual task adaptation benchmark. arXiv

Zhao H, Yan Z, Torresani L, & Torralba A (2019) HACS: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*

Zhou B, Andonian A, Oliva A, & Torralba A (2018) Temporal relational reasoning in videos. *ECCV*

Zhou L, Kalantidis Y, Chen X, Corso JJ, & Rohrbach M (2019) Grounded video description. In *CVPR*

Zhou B, Krähenbühl P, & Koltun V (2019) Does computer vision matter for action? Science Robotics

Zhou L, Xu C, & Corso JJ (2017) Towards automatic learning of procedures from web instructional videos. In *AAAI*

Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, & Torralba A (2017) Scene parsing through ADE20K dataset. In *CVPR*