BURST: A Benchmark for Unifying Object Recognition, Segmentation and Tracking in Video

Ali Athar[1] Jonathon Luiten[1,2] Paul Voigtlaender[3] Tarasha Khurana[2] Achal Dave[4]
Bastian Leibe[1] Deva Ramanan[2]

[1] RWTH Aachen University, Germany [2] Carnegie Mellon University, USA [3] Google [4] Amazon

{athar,luiten,leibe}@vision.rwth-aachen.de {tkhurana,deva}@cs.cmu.edu
voigtlaender@google.com achald@amazon.com

Abstract

Multiple existing benchmarks involve tracking and segmenting objects in video e.g., Video Object Segmentation (VOS) and Multi-Object Tracking and Segmentation (MOTS), but there is little interaction between them due to the use of disparate benchmark datasets and metrics (e.g. $\mathcal{J}\&\mathcal{F}$, mAP, sMOTSA). As a result, published works usually target a particular benchmark, and are not easily comparable to each another. We believe that the development of generalized methods that can tackle multiple tasks requires greater cohesion among these research sub-communities. In this paper, we aim to facilitate this by proposing BURST, a dataset which contains thousands of diverse videos with high-quality object masks, and an associated benchmark with six tasks involving object tracking and segmentation in video. All tasks are evaluated using the same data and comparable metrics, which enables researchers to consider them in unison, and hence, more effectively pool knowledge from different methods across different tasks. Additionally, we demonstrate several baselines for all tasks and show that approaches for one task can be applied to another with a quantifiable and explainable performance difference. Dataset annotations are available at: `https://github.com/Ali2500/BURST-benchmark`.

1. Introduction

Segmenting and tracking multiple objects in video is widely researched because of applications in autonomous robots and self-driving vehicles. Over time however, this broadly defined task has splintered into multiple datasets and benchmarks, each with its own sub-community. Even though tasks such as Video Object Segmentation (VOS) and Multi-Object Tracking and Segmentation (MOTS) are closely related, there is a lack of interaction between their sub-communities.

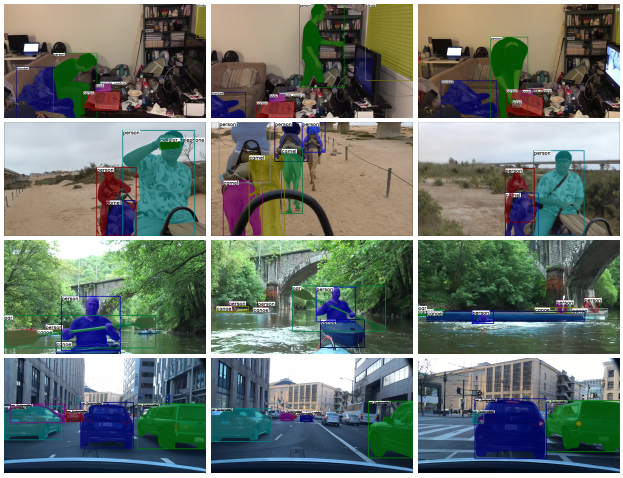

Figure 1. Assorted object annotations from BURST showing diverse outdoor, indoor and driving scenes.

Our work aims to remedy this; we propose BURST: a dataset containing a large, diverse set of videos with object mask annotations, and an associated benchmark with six related tasks. BURST is based on the existing TAO dataset by Dave *et al*. [5] for bounding-box level multi-object tracking, but has been extensively re-annotated with pixel-precise masks. The videos in our dataset include indoor and outdoor scenes, 'in-the-wild' videos, scripted movie scenes, and street scenes captured from moving vehicles. Examples can be seen in Fig. 1. The six tasks in our benchmark are organized into a hierarchical taxonomy which is illustrated in Fig. 2. All tasks fall under the umbrella of requiring pixel-precise segmentation and tracking of potentially multiple objects in video sequences.

The first level in our task hierarchy splits tasks based on the set of target objects that have to be segmented/tracked. For *exemplar-guided* tasks, an explicit cue is given for each of the target objects. For *class-guided* tasks, the set of target objects are all those which belong to a predefined set

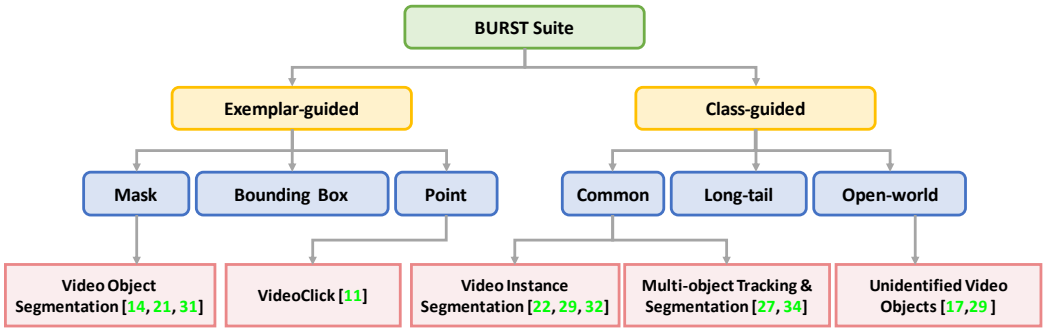

Figure 2. **BURST Task Taxonomy/Hierarchy.** Boxes in the bottom row give examples of existing benchmarks which tackle that task.

of object classes. The exemplar-guided stream is further divided into three tasks where, for the first video frame in which the target object appears, we are given either (i) the object mask, (ii) its bounding box or (iii) a random point inside the object. The class guided stream is also further divided into three tasks where the pre-defined class set is either (i) a small set of common object classes, (ii) a larger set of classes with several infrequently occurring classes (*i.e.* long-tail), or (iii) an 'open-world' task [17] where methods are trained on a small set of known classes, but during inference are expected to additionally track and segment objects belonging to a larger, previously unseen set of classes.

Fig. 2 (bottom row) shows which existing benchmarks map onto our task taxonomy *e.g.* Video Object Segmentation [21, 31] is identical to our mask exemplar-guided task, whereas Video Instance Segmentation (VIS) [32, 22] and Multi-Object Tracking and Segmentation (MOTS) [27] are similar to the common class-guided task.

The hierarchy shows that these tasks are highly related to one another. As we will show in Sec. 7, research advances targeting one task can be utilized for other tasks. For example, state-of-the-art methods [4, 33, 20, 23] for exemplar-guided tasks work by 'propagating' object masks from one video frame to another. Here we note that improvements in mask-propagation (*i.e.* temporal association) can benefit class-guided methods. With BURST, we aim to bring together methods for these tasks under a single umbrella benchmark to encourage more knowledge exchange. To further facilitate unification and interaction, we use the same set of metrics based on Higher-Order Tracking Accuracy (HOTA) [18] for all tasks. This enables direct, quantitative comparison between different methods targeting different tasks. To demonstrate the usefulness of this feature, we setup several effective baselines for our proposed tasks, some of which are constructed by bootstrapping standard approaches for other tasks. The comparability of the resulting scores offers interesting insight into how well methods generalize across tasks.

To summarize, we propose BURST: a large, diverse and challenging dataset with mask-level object annotations, and an associated benchmark with 6 tasks related to segmenting and tracking multiple objects in video. Methods can be evaluated for one or more tasks using the same underlying data and comparable metrics. This aims to encourage greater cohesion and knowledge exchange between researchers working on these tasks, and accelerate development of generalized methods that can tackle multiple tasks.

2. The BURST Benchmark

Existing object tracking and segmentation datasets are typically geared towards certain types of video scenes *e.g.* in-the-wild internet videos [21, 32, 31, 22], outdoor street scenes captured from a driving vehicle [7, 34, 2, 11]. The videos in BURST, on the other hand, cover multiple types of scenes and encompasses a large set of 482 object classes. We use the videos from TAO [5], which is in-turn composed from videos belonging to 7 different datasets: ArgoVerse [2] and BDD [34], which contain outdoor driving scenes captured from moving vehicles, LaSOT [6] and YFCC100M [26], which contain assorted, in-the-wild videos from the internet, and AVA [8], Charades [24] and HACS [35], which contain videos with human-human and human-object interactions, but with some subtle differences: Charades contains mostly indoor scenes with slow object motion, AVA contains snippets from scripted movies, and HACS contains in-the-wild internet videos. We refer the reader to Fig. 1 from Dave *et al.* [5] for an illustrated overview of our dataset classes.

BURST contains 2,914 videos with a lower frame dimensions of at least 480px. The videos are ∼30s in length, and the training, validation and test set contain 500, 993, and 1421 videos, respectively. The training set is annotated at 6fps whereas both validation and test sets are annotated at 1fps. Table 1 summarizes statistics for BURST.

Federated Annotations. Similar to TAO [5], the annotations in BURST are *federated*, *i.e.* not all objects belonging to the predefined set of object classes are annotated in every video. This is similar to the philosophy behind the image-level LVIS [9] and OpenImages [15] datasets. Every video in BURST contains the following, in addition to the non-exhaustive annotations: (1) a list of object classes which are non-exhaustively annotated, and (2) a list of object classes

Table 1. Statistics for BURST train, validation and test sets.

	Train	Validation	Test	Total
Annotation fps	6	1	1	-
Videos	500	993	1421	2,914
Total video length (hrs)	4.94	9.84	14.12	28.9
Object tracks	2,645	5,481	7,963	16,089
Annotated frames	107,144	36,375	52,194	195,713
Object masks	318,200[1]	114,825	167,132	600,157

[1] includes 212,477 automatically generated and consistency-verified masks

which are not present in the video. This information enables us to derive three sets of videos for every object class: where it is present, absent and non-exhaustively present. This in-turn is used to penalize false positives and negatives for each object class during evaluation. We refer readers to the TAO dataset paper [5] for more details.

3. Comparison to Related Datasets

There exist several datasets of various sizes which tackle one or more tasks evaluated by our benchmark. Tables 2 and 3 compare BURST to these existing datasets.

3.1. Comparison by Tasks

Table 3 shows which tasks each dataset/benchmark evaluates in terms of our task taxonomy (Fig. 2). We note that existing benchmarks typically address one or at most two tasks. The '/' entries mean that the dataset does not evaluate the given task, but that it is possible to do so *e.g*. a class-guided benchmark can also be formulated as an exemplar-guided benchmark by assuming that the first-frame object masks are known during inference. For exemplar-guided tasks, the two most common benchmarks are DAVIS [21] and YouTube-VOS [31]. Both contain diverse, in-the-wild, videos from the internet which are ∼5-10s in length. VOT [14] contains longer videos, but is a single-object tracking dataset.

Looking at the class-guided stream, most datasets can be assigned to one of two distinct groups. On one hand, benchmarks such as BDD [34], KITTI [7] and MOTS-Challenge [27] are inspired from classical Multi-Object Tracking (MOT). They target autonomous driving problems, and contain lengthy videos of street scenes captured from a driving vehicle or a walking pedestrian. We see that KITTI and MOTS-Challenge evaluate methods using 'sMOTSA', which is an extension of the popular MOTA (Multi-Object Tracking Accuracy) measure [25] for when segmentation masks are used instead of bounding-boxes. On the other hand, datasets like YouTube-VIS [32] and UVO [29] appear more related to Video Object Segmentation (VOS), and typically contain diverse, but shorter videos from the internet. OVIS [22] can be seen as an extension of YouTube-VIS with longer videos and more object occlusion. UVO stands out from the others in that it could be used for the open-world task since it contains mask annotations for a anything which humans would consider to be 'objects' as opposed to being restricted to a small class of objects. Datasets in this category use mean Average Precision (mAP) as an evaluation measure. These benchmarks can thus be seen as video extensions of image-level instance segmentation benchmarks such as COCO [16], LVIS [9] and OpenImages [15] where mAP is the metric of choice.

In contrast to all of the above, BURST contains the required annotations to evaluate all six tasks. In particular, the long-tail open-world tasks are enabled by the fact that our object class set is sufficiently large.

3.2. Comparison by Difficulty

Table 2 lists several datasets along with various parameters which subjectively determine their 'difficulty'. In terms of video length, the average sequence in BURST lasts 36.8s, which is longer than the other datasets. Lengthy videos are challenging because of more object instances, longer occlusions, and are also more memory demanding because of the higher frame count. In terms of number of annotations, BURST contains a total of ∼600k object masks across ∼200k video frames, which is larger than most other datasets except BDD and UVO. With regard to object classes, BURST contains objects belonging to 482 possible classes, which is significantly higher than the class set for other benchmarks. To the best of our knowledge, we are the first to provide pixel-precise object annotations for such a large set of object classes. Besides increasing object diversity, this feature also enables us to evaluate methods for the long-tail class-guided task. As mentioned in Sec. 2 however, our annotations are federated, *i.e*. not exhaustive. Finally, we note that that BURST can better evaluate the generalization capability of methods since it contains more scene diversity compared to existing datasets, many of which focus on specific settings *e.g*. driving scenes.

4. Dataset Creation

We build upon the TAO dataset [5] which contains object bounding-box annotations at 1fps. We professionally re-annotated this to obtain pixel-precise masks for all 342,052 object bounding-boxes[1]. We then set out to increase the temporal density of annotations in the training set from 1fps to 6fps. Visualizing a sequence of annotations at 1fps shows large movements and appearance changes between successive frames. It is challenging to train tracking-related methods on this data since they are designed to learn video motion cues from smooth video frame progression. However, annotating object masks at the full video frame-rate (24-30fps) would be in-feasibly expensive and highly redundant since there is usually little scene change between successive frames. Annotating at 6fps is thus a compromise (also used by other datasets [21, 31, 32]) since it reduces anno-

[1] We re-used the small set of 27,500 masks published by [28].

Table 2. **Dataset Comparison By Size and Difficulty.** Comparison of datasets according to various measures of 'difficulty'. Statistics for validation/test may not be exact if data is not publicly available.

	Difficulty				Train Size				Validation / Test Size			
Dataset	Setting	Length (hrs)	Masks / Frame	# Classes	Ann Masks	Ann Tracks	Ann Frames	Ann Vids	Ann Masks	Ann Tracks	Ann Frames	Ann Vids
VOT [14]	Single Object	10.7	1	-	0	0	0	0	19,903	62	19,903	62
DAVIS'17 [21]	Internet Videos	2.9	2.6	78	10,238	144	4,219	60	16,841	242	6,240	90
YT-VOS [31]	Internet Videos	4.5	1.63	94	12,918	6,459	94,440	3,471	4,310	2,155	28,825	1,048
BDD [34]	Driving	40	11.4	7	347,442	17,838	30,745	154	77,389	4,873	6,475	32
KITTI-MOTS [27]	Driving	39.0	5.2	2	38,197	748	8,008	21	61,906	961	11,095	28
MOTS-Chal. [27]	Surveillance	34.4	10.0	1	26,894	228	2,862	4	32,269	328	3044	4
YT-VIS [32]	Internet Videos	4.5	1.69	40	103,424	3,774	61,845	2,238	29,431	1,092	17,415	645
UVO [29]	Human Actions	3	12.3	-	416,001	76,627	39,174	5,641	177,153	28,271	18,966	5,587
BURST	General / Diverse	28.9	3.1	482	318,200	2,645	107,144	500	281,957	13,444	88,569	2,414

Table 3. **Dataset Comparison by Task.** '/' means that the dataset contains annotations to setup the given task, but this is not done officially as part of that benchmark.

	Exemplar-guided			Class-guided		
	Mask	Box	Point	Common	Long-Tail	Open-World
VOT [14]	✓	✓	/	✗	✗	✗
DAVIS [21]	✓	/	/	✗	✗	✗
YouTube-VOS [31]	✓	/	/	✗	✗	✗
BDD [34]	/	/	/	✓	✗	✗
KITTI-MOTS [27]	/	/	/	✓	✗	✗
MOTS-Chal. [27]	/	/	/	✓	✗	✗
YouTube-VIS [32]	/	/	/	✓	✗	✗
OVIS [22]	/	/	/	✓	✗	✗
UVO [29]	/	/	/	✓	✗	✓
BURST	✓	✓	✓	✓	✓	✓

tation cost while still ensuring smooth scene progression. For BURST however, even 6fps annotations would require 255,654 additional mask annotations for the training set. To reduce cost and human effort, we instead developed a semi-automatic procedure to do this, as explained below:

1. Automatic Mask Propagation Interestingly, the task of temporally densifying annotations is practically identical to the mask exemplar guided task referred to in Fig. 2: given an object mask in a certain frame, we need masks for the same object in other video frames. In this case, the video length over which this mask propagation step has to be performed is quite short – at most 1s since we already have human-labeled 1fps annotations. We found that two recent, state-of-the-art methods for 'Video Object Segmentation', namely STCN [4] and AOT-L [33], perform this task quite well with off-the-shelf trained weights. To further improve mask quality, we obtain two different sets of results from each of these methods by running them in two different ways. As shown in Fig. 3 (left), given a pair of annotated frames with a number of non-annotated frames in between, we can either run the method with the first frame as the reference and propagate *forward* sequentially, or, starting from the last frame as reference, propagate *backward* sequentially. Doing so for both methods results in a total of 4 different sets of propagated masks. Additionally, we use STCN to obtain a fifth, tie-breaking *two-sided* result by using both annotated frames as the reference frames (Fig. 3, right), and propagating the masks directly to each of the other non-annotated frames (*i.e*. the frame history update mechanism in STCN is disabled). Thus, we have a total of 5 masks for each object. We subsequently perform a per-pixel majority vote to obtain a final *consensus mask*.

2. Mask Quality Assessment Although most annotations produced by step 1 are high quality, there are several failure cases *e.g*. poor lighting, occluded scenes, erratic camera motion. To identify them, one could manually inspect each object mask and decide if it is of ground-truth quality. Even though doing this is still significantly less costly than fully annotating the object masks, we nonetheless developed a more efficient yet effective procedure for evaluating mask quality: we compute the IoU of each of the five masks generated in step 1 with the *consensus mask*. The five resulting IoUs are then averaged to obtain a final metric in $[0, 1]$ which is treated as a quality score Q for the *consensus mask*.

3. Manual Re-annotation of Low Quality Masks. To decide which masks are of sub-standard quality, we consider two key measures: the score Q from step 2, and the pixel mask area of the consensus mask. We asked two professional annotators to manually assess the quality of a set of 250 object masks. These were sampled such that they are uniformly distributed with respect to their Q scores and pixel areas. The annotators were asked to assign one of three ratings to each object mask: (1) 'good': mask quality is as good as human annotated ground-truth, (2) 'satisfactory': there are visible errors *e.g*. object contours are imperfect, minor instances of mask fragmentation, but overall still acceptable and (3) 'bad': there are unacceptable errors *e.g*. object ID switches, gross under/over-segmentation. Fig. 4 illustrates the results of this survey: each object mask is shown as a point whose color reflects the human-assigned rating. The points are plotted w.r.t their Q score and mask pixel areas. We observe a strong correlation between both of these measures and the human-perceived mask quality, since most 'bad' masks are in the lower-left corner of the plot, and vice versa. Based on this plot, we decided to manually re-annotate all object masks whose Q scores were below 0.8, or whose mask areas were smaller than 750 pixels. This region is highlighted in red in Fig. 4.

By using this workflow to densify the training set an-

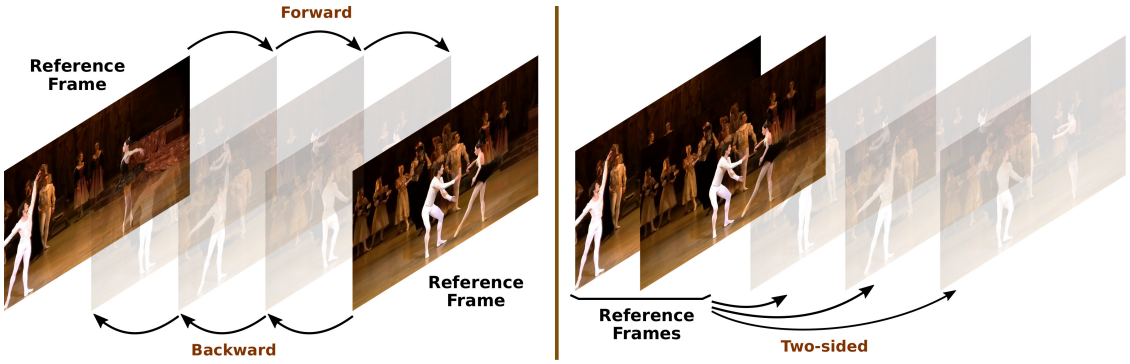

Figure 3. Illustration of mask propagation techniques for densifying the training set. STCN [4] and AOT-L [33] are both executed in the forward and backward settings. The two-sided setting is only executed for STCN.

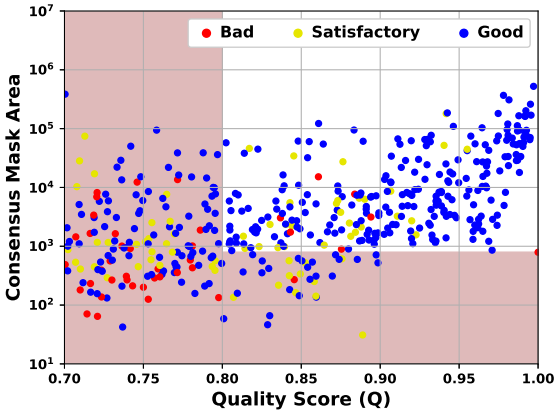

Figure 4. Human assessment of mask quality plotted against our automated quality score Q and pixel area of the consensus masks. All masks falling in the red region were manually re-annotated.

notations from 1fps to 6fps, we only require 43,177 out of 255,654 object masks (16.9%) to be manually annotated since the remaining automatically generated masks passed the quality threshold to be considered ground-truth.

5. BURST Task Taxonomy

As explained in Sec. 1, several existing benchmarks involve closely related tasks pertaining to segmenting and tracking multiple objects in video, but there is limited cross-interaction between their respective research sub-communities. With BURST, we aim to unify these disparate benchmarks under a single umbrella with shared data and consistent evaluation metrics. The six tasks constituting BURST are illustrated in Fig. 2 and explained below.

5.1. Exemplar-guided

This set of tasks requires tracking and segmenting multiple target objects in a video given some ground-truth cue for each of these objects in the first video frame in which they appear. Note that this may not necessarily be the first frame of the video. The three tasks in this stream are based on the type of the given cue:

1) Mask. The method is given the segmentation mask for each of the target objects in the first frame.

2) Box. The bounding box coordinates of the target objects in the first frame are given. Note that the predicted output should still be pixel-precise segmentation masks.

3) Point. This is the most challenging of the three where the method is only given one pixel coordinate which lies inside the target object mask. Again, the predicted output should still contain segmentation masks.

5.2. Class-guided

For this set of tasks, methods are required to track, segment and assign a class label to all objects in a video which belong a pre-defined set of object classes. The three tasks in this stream are:

4) Common. Here, the target class set includes 78 classes from the popular COCO dataset [16] spanning diverse object categories *e.g.* animals, persons, vehicles, furniture, food items.

5) Long-tailed. This task involves a large set of 482 object classes from the LVIS dataset [9]. It is challenging because several classes contain very few training samples.

6) Open-world. The idea behind open-world instance segmentation [17] is that methods are trained on a certain, 'known' set of object classes, but during inference, they are expected to additionally segment objects belonging to an 'unknown' class set. Methods need not assign class labels to the predicted instances, and the evaluation does not penalize false positives. For our open-world task, the 78 'common' classes are the 'known' set, and the 'unknown' set includes everything that is in the 482 class 'long-tail' set, but not in the 'common' set.

6. Unified Evaluation Metrics

We evaluate all tasks using Higher Order Tracking Accuracy (HOTA) [18] because it strikes a good balance between measuring frame-level detection and temporal association accuracy. For the open-world task, a slightly modified,

recall-based variant of HOTA called Open World Tracking Accuracy (OWTA) is used.

HOTA. To calculate HOTA [18], the predicted detections (per-frame) are first matched to the ground truth detections based on the IoU between their masks. Using this mapping, the Detection Accuracy (DetA) and the Association Accuracy (AssA) can be calculated and combined by taking their geometric mean to obtain the HOTA score, *i.e*.

$$\mathrm{HOTA} = \sqrt{\mathrm{DetA} \cdot \mathrm{AssA}}. \tag{1}$$

Multiple IoU thresholds are used to compute the prediction $\leftrightarrow$ ground-truth matching; the final HOTA (and DetA, AssA) is calculated by averaging over the thresholds.

Detection Accuracy (DetA). Using the mapping between predicted and ground truth detections, these detections can be partitioned into a set of True Positive (TP), False Positive (FP), and False Negative (FN) detections. The DetA, which solely measures the quality of the detections while disregarding track associations, can then be obtained by

$$\mathrm{DetA} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}| + |\mathrm{FP}|}. \tag{2}$$

Association Accuracy (AssA). To calculate the AssA, an association score $\mathcal{A}(c)$ is calculated for each true positive detection c. The final AssA score is obtained by averaging over the set of true positive detections TP:

$$\mathrm{AssA} = \frac{1}{|\mathrm{TP}|} \sum_{c \in \{\mathrm{TP}\}} \mathcal{A}(c). \tag{3}$$

The association score $\mathcal{A}(c)$ for true positive detection c is calculated as

$$\mathcal{A}(c) = \frac{|\mathrm{TPA}(c)|}{|\mathrm{TPA}(c)| + |\mathrm{FNA}(c)| + |\mathrm{FPA}(c)|}, \tag{4}$$

where True Positive Associations (TPAs), False Positive Assocations (FPAs), and False Negative Associations (FNAs) [18] are computed by comparing the whole predicted track which goes through detection c with the whole ground truth track which goes through detection c. We refer the reader to Luiten *et al*. [18] for detailed explanations.

Object Classes. To handle multiple object classes, HOTA can be calculated separately for each class, followed by an averaging step to yield a final metric. To facilitate easier performance analysis over different object classes, we average the per-class HOTA scores over three different sets of object classes: (1) 'common' set, which contains 78 object classes from COCO [16], (2) 'uncommon' set, which contains 404 infrequently occurring object classes from LVIS [9], and (3) 'all' set, which is the union of both ($78 + 404 = 482$ classes). We denote these three metrics with $\mathrm{HOTA}_{\mathrm{com}}$, $\mathrm{HOTA}_{\mathrm{unc}}$ and $\mathrm{HOTA}_{\mathrm{all}}$, respectively.

HOTA for Exemplar-guided Tasks. The evaluation for the exemplar-guided task is identical to the class guided task, and the scores can be directly compared. However, it should be noted that the exemplar-guided methods inherently receive extra ground-truth information: a mask/box/point, and the class label for each target object.

6.1. Open-world Evaluation

For the open-world task, methods are expected to segment and track objects of previously unseen classes. Since it is infeasible to label every single object (even the definition of 'object' is ambiguous), we have to assume that the prediction may contain valid objects which are not covered by the ground truth. This entails that false positive detections should not be penalized, and hence, for the open-world task, we replace HOTA with Open-World Tracking Accuracy (OWTA) [17], which is calculated as:

$$\mathrm{OWTA} = \sqrt{\mathrm{DetRe} \cdot \mathrm{AssA}}, \tag{5}$$

where the Detection Recall (DetRe) is given by

$$\mathrm{DetRe} = \frac{|\mathrm{TP}|}{|\mathrm{TP}| + |\mathrm{FN}|}, \tag{6}$$

Note that DetRe is similar to DetA, but it disregards false positives (FP). To prevent methods from obtaining a high score by simply predicting an extremely large number of detections, we mandate that object mask predictions for the open-world task cannot overlap with each other.

7. Baselines

For each of our six tasks (Sec. 5), we implement baselines utilizing existing works with off-the-shelf trained models. These serve a point of comparison for future works, and also show how one approach can be utilized for multiple tasks, and how performances across tasks can be directly compared and analyzed. In general, we utilize image-level object detectors in the context of 'tracking-by-detection', where the tracking task is conceptually divided into two steps: a 'detection' step in which objects are segmented in individual frames, followed by a 'tracking' step in which the per-frame detections are associated over time. Although recent state-of-the-art methods [1, 19, 30, 3] diverge from this paradigm by jointly segmenting and tracking objects in video clips, we find that it remains a versatile approach for tackling the tasks in BURST. We construct functional baselines for each task using some variation of tracking-by-detection. The following sub-sections detail each of the baselines and the results are presented in Table 4.

7.1. Exemplar-guided

We show two baselines for each task in this stream: (1) applying STCN [4], which is a recent 'Video Object Segmentation' method for propagating object masks through

Table 4. Baseline results for all tasks using various methods. Evaluation metrics are reported separately for 'common', 'uncommon' and 'all' classes. Object detector training data: *: COCO, †: LVIS.

			Validation			Test		
		Baseline Method	HOTA_{all}	HOTA_{com}	HOTA_{unc}	HOTA_{all}	HOTA_{com}	HOTA_{unc}
Exemplar Guided	Mask	STCN [4]	49.8	52.2	49.2	52.4	51.1	52.7
		Box Tracker* [12]	18.0	35.8	13.6	14.1	28.0	11.4
	Box	STCN (PointRend)	45.2	48.9	44.3	46.0	48.9	45.4
		STCN (Matched Det*)	24.5	47.6	18.7	25.0	41.9	21.7
		Box Tracker*	13.7	34.2	8.6	13.6	27.7	10.8
	Point	STCN (Matched Det*)	24.4	44.0	19.5	24.9	39.5	22.0
		Box Tracker*	12.7	31.7	7.9	10.1	24.4	7.3
Class Guided	Common	STCN Tracker*	-	51.2	-	-	34.6	-
		Box Tracker*	-	45.5	-	-	34.3	-
	Long-tail	STCN Tracker†	5.5	17.5	2.5	4.5	17.1	2.0
		Box Tracker†	8.2	27.0	3.6	5.7	20.1	2.9
			OWTA_{all}	OWTA_{com}	OWTA_{unc}	OWTA_{all}	OWTA_{com}	OWTA_{unc}
	Open-world	STCN Tracker	64.6	71.0	25.0	57.5	62.9	23.9
		Box Tracker	60.9	66.9	24.0	55.9	61.0	24.6
		OWTB [17]	55.8	59.8	38.8	56.0	59.9	38.3

video, and (2) a simple box tracker which builds object tracks by starting from the given first-frame mask, and then associating object detections in future frames using Hungarian matching based on their bounding-box overlap.

Mask-guided. Here, STCN consistently out-performs the box tracker for both class sets since it is a state-of-the-art method for exemplar-guided tracking whereas the box tracker is a basic approach. Note how the difference widens for uncommon classes where STCN achieves 49.2 HOTA_{unc} (validation) whereas box tracker only achieves 13.6. This is because STCN is class-agnostic, and can track any given first-frame object mask, whereas the box tracker uses object detections produced by a Mask2Former [3] model trained on COCO [16] (*i.e*. 'common' classes). Nonetheless, the applicability of tracking-by-detection for this task shows that exemplar-guided tasks may benefit from future improvements to image-level detectors.

Box-guided. For the box and point guided tasks, we formulate baselines by treating them as extensions of the mask-guided task with an additional 'box $\rightarrow$ mask' or 'point $\rightarrow$ mask' pre-processing step which regresses the segmentation mask from the given first-frame bounding-box or point, respectively. For the box-guided task, we do this in two ways: (1) We compute the IoU between the given bounding box with all the bounding boxes of the image detections for that frame, and assign the mask belonging to the detection with the highest overlap, and (2) we input the given first-frame bounding box to a PointRend [13] based mask regression head from a MaskRCNN [10] model, and use the resulting segmentation mask. Looking at Table 4, we see that the scores for box-guided are generally lower than those for mask-guided due to the additional 'box $\rightarrow$ mask' regression step. Among box-guided scores, PointRend performs much better than using the best-matched detection. For HOTA_{unc} in particular, the PointRend baseline achieves 44.3 (validation) whereas the matched detection baseline only gets 18.7. Given the fact that both networks (the image detector used for matching, and the PointRend mask head) are trained on COCO, it shows that PointRend is a much more robust 'box $\rightarrow$ mask' regressor than matching detections.

Point-guided. Finally, for the point-guided task we implement 'point $\rightarrow$ mask' by taking the mask for the highest scoring detection which contains the given point. Since this technique is prone to errors, we see that in terms of all three metrics, the baselines for the point-guided task are worse compared to those for box-guided and mask-guided.

7.2. Class-guided

We show two tracking-by-detection baselines per task: (1) A simple box tracker which links per-frame object detections using box IoU followed by Hungarian matching, and (2) an 'STCN tracker', where, in order to associate object detection masks in frame t with those in $t+1$, we use STCN to propagate the masks from frame $t+1$ to frame t, and then use the IoU between these propagated masks and the object masks in frame t as an association metric.

Common. This task requires segmenting and tracking objects belonging to the 78-class 'common' set. Here, STCN tracker performs better than the box tracker (51.2 vs. 45.5 HOTA_{com}) because STCN-based mask propagation is more accurate for temporal association compared to bounding-box IoU. By effectively utilizing STCN, a method designed to tackle the mask exemplar-guided task, for class-guided tracking, we exemplify the related nature of these tasks and the potential for knowledge exchange among them. Note that we do not evaluate the common task for 'uncommon'

classes since this is not required by the task definition.

Long-tail. Here, methods are required to segment and track objects belonging to the 482-class 'all' set. We note that the scores here are significantly worse than those for the common task. This is because we used a MaskRCNN [10] model trained on LVIS [9] to obtain object detections for this larger set of classes. We observed that the detections produced by this network are of poor quality. Even for the 'common' classes, the performance of both baselines reduces drastically when these detections are used (27.0 HOTA_{com} on validation for box tracker vs. 45.5 for the common task). Also note that here, the STCN tracker performs worse than the box tracker, even though the opposite was true for the common task. The reason is that STCN performs erroneous mask propagation when the input mask quality is bad. Hence, in this case the more basic bounding-box IoU tracker performs comparatively better. To the best of our knowledge, this is the first time that a quantitative comparison of video object tracking methods is given in terms of their performance on such a large class set. We hope that our benchmark will encourage other researchers to discover ways of mitigating this large performance gap.

Open-world. Finally, for the open-world task, we use the OWTA metric which is similar to HOTA, but without penalization for false positives. As per the task definition, methods can only be trained on the 'common' class set, but during inference, are expected to additionally segment objects belonging to the 'uncommon' set. Here, we again use the box tracker and STCN tracker with image-level detections from a Mask2Former [3] model trained on COCO. We additionally report results for the baseline proposed by Liu *et al*. [17] (OWTB). Unsuprisingly, all methods suffer performance degradation for the 'uncommon' set. The STCN tracker achieves the highest HOTA_{all} score (64.6 on validation), but OWTB performs significantly better in terms of HOTA_{unc} (38.8) compared to the next best baseline (box tracker: 25.0).

7.3. Comparison Across Tasks

Using consistent metrics enables us to directly compare results for different methods across different tasks. We illustrate this comparison for our baselines in Fig. 5 which charts the best-performing baseline on the validation set for each task. Note that we omitted open-world results since the OWTA metric differs slightly from HOTA. For HOTA_{com}, the mask exemplar-guided score of 52.2 is only slightly higher than that for the common class-guided task (51.2). This may seem surprising since the exemplar-guided task is inherently easier because, for each target object, methods have access to (1) an explicit cue (mask/box/point), and (2) the class label. However, we noticed that exemplar-guided methods often lose the target object (*e.g*. due to occlusion or erratic motion), and thereafter cannot recover it; for both

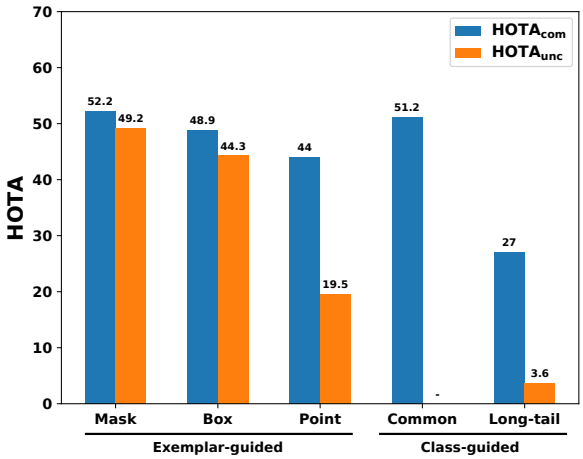

Figure 5. Comparison of baseline performances for different tasks.

STCN and box tracker, if the object is not track-able for more than a certain number of frames, it is assumed to have disappeared. On the other hand, class-guided methods predict an arbitrary number of tracks which may collectively cover more of the ground-truth object. Because HOTA calculation involves Detection Accuracy (DetA), the class-guided methods receive partial credit for correct per-frame detections even if the object ID is inconsistent/fragmented over time. This effect is more visible for BURST because it contains longer videos (∼30s) compared to existing exemplar-guided benchmarks [21, 31] (∼5-10s).

For HOTA_{unc} however, even the worst exemplar-guided score (19.5) is much higher than the 3.6 achieved for class-guided, because exemplar-guided methods are inherently class-agnostic and can propagate arbitrary object masks, but our class-guided tracking-by-detection baselines are very sensitive to per-frame object detection/classification quality, which is currently quite poor for this larger class set.

8. Conclusion

We present BURST: a benchmark which unifies six tasks related to object recognition, segmentation and tracking in video with a clear task taxonomy and consistent evaluation metrics. Our dataset contains a large and diverse video set with pixel-precise masks for a large vocabulary of object classes. We temporally densified the object masks for the training set using a semi-automated pipeline which yields accurate results while drastically reducing human annotation effort. Finally, we presented a number of baselines for the proposed tasks and analyzed their performance. We hope that our benchmark will serve as a valuable resource for researchers to evaluate their object tracking methods.

Acknowledgements. This project was partially funded by ERC Consolidator Grant DeeVise (ERC-2017-COG-773161) and the CMU Argo AI Center for Autonomous Vehicle Research.

References

[1] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *ECCV*, 2020.

[2] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019.

[3] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *CVPR*, 2022.

[4] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021.

[5] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *ECCV*, 2020.

[6] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[8] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[9] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[11] Namdar Homayounfar, Justin Liang, Wei-Chiu Ma, and Raquel Urtasun. Videoclick: Video object segmentation with a single click. *arXiv preprint arXiv:2101.06545*, 2021.

[12] Arne Hoffhues Jonathon Luiten. Trackeval. `https://github.com/JonathonLuiten/TrackEval`, 2020.

[13] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020.

[14] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka ˇCehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *ECCV*, 2018.

[15] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 2020.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014.

[17] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Deva Ramanan, Bastian Leibe, Aljoša Ošep, and Laura Leal-Taixé. Opening up open world tracking. In *CVPR*, 2022.

[18] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 2020.

[19] Sabarinath Mahadevan, Ali Athar, Aljoša Ošep, Sebastian Hennen, Laura Leal-Taixé, and Bastian Leibe. Making a case for 3d convolutions for object segmentation in videos. *BMVC*, 2020.

[20] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019.

[21] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS Challenge on Video Object Segmentation. *arXi*, 2017.

[22] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation. *arXiv*, 2021.

[23] Hongje Seong, Seoung Wug Oh, Joon-Young Lee, Seongwon Lee, Suhyeon Lee, and Euntai Kim. Hierarchical memory matching network for video object segmentation. In *PICCV*, 2021.

[24] Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, Ivan Laptev, and Abhinav Kumar Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ECCV*, 2016.

[25] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, R Travis Rose, Martial Michel, and John Garofolo. The clear 2007 evaluation. In *Multimodal Technologies for Perception of Humans*. 2007.

[26] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016.

[27] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019.

[28] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. In *WACV*, 2021.

[29] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021.

[30] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021.

[31] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark. *arXiv*, 2018.

[32] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019.

[33] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021.

[34] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

[35] Hang Zhao, Zhicheng Yan, Lorenzo Torresani, and Antonio Torralba. Hacs: Human action clips and segments dataset for recognition and temporal localization. *arXiv*, 2019.