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

Supplementary Material

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

1. Dataset Visualization

The supplementary zip file contains a dataset viewer, for viewing all of the videos and annotations in the BURST dataset. Please refer to the 'readme' file for further instructions. A screenshot from this viewer, showing examples of some video frames with annotated masks from our training set is shown in Fig. 1.

2. Extended Baseline Results

Tables 1 and 2 show extended results for our baselines (Sec. 7 in main text) for the validation and test set, respectively. Tables 3 and 4 show extended results for the open-world class-guided task for the validation and test sets, respectively. Here, aside from the HOTA score, we also provide the DetA, AssA and mAP scores. Additionally, we provide another variant of HOTA called $\mathrm{HOTA}^{\mathrm{obj}}$ where the final score is calculated by averaging the per-object HOTA scores, instead of averaging over the object classes. We draw the following observations and comparisons from the tabulated results:

2.1. HOTA vs. mAP

mAP (mean average precision) is used as a metric by several existing benchmarks related to video object tracking and segmentation [13, 11, 3]. It works by computing the mask IoU at the track-level (*i.e.* across the whole video) and then uses Hungarian matching to assign at most one predicted track to each ground-truth track. All other predicted tracks are considered as false positives, even if they intersect strongly with a ground-truth track. On the other hand, HOTA gives weighting to both per-frame detection accuracy (DetA) as well as temporal association accuracy (AssA).

This difference can be noted by comparing the mask exemplar-guided results with those for the common class-guided task. We see in Table 1 that in terms of HOTA on common classes, STCN achieves 52.2 which is only slightly higher than the 51.2 achieved by the STCN Tracker. However, in terms of mAP, the difference is much larger (27.5 vs. 13.6). This is because the mask exemplar-guided task provides the first-frame ground-truth mask for every object during inference. This often results in one predicted track having reasonably good overlap with the ground-truth. However, once the target object is lost due to tracking errors, it can no longer be recovered. This results in a higher mAP score because there is usually one good match in the predictions for each ground-truth track, and no false positives. For the class-guided task however, the first-frame mask is not given, and methods predict an arbitrary number of tracks which may collectively capture a certain ground-truth object (in multi-object tracking parlance, we would say that the predicted tracks are *fragmented*). This results in a low mAP score because there is no single high-quality predicted track which has a high IoU with the ground-truth, and every predicted track aside from the best-matched one is considered as a false positive.

With HOTA however, we can quantitatively analyze this phenomenon (Table 1): in terms of DetA, the exemplar-guided STCN achieves 44.7 which is lower than the 51.7 for the class-guided STCN Tracker. This shows that in terms of per-frame detections, the class-guided method correctly predicts more of the ground-truth. In terms of AssA however, STCN achieves 62.5 which is higher than the 51.9 for the STCN Tracker. This shows that temporal association quality is better for the exemplar-guided method (as we hy-

Table 1. Extended baseline results for the validation set with multiple metrics. Evaluation metrics are reported separately for ‘common’, ‘uncommon’ and ‘all’ classes. Object detector training data: *: COCO, †: LVIS.

		Baseline Method	All					Common					Uncommon				
			DetA	AssA	HOTA	HOTA^{obj}	mAP	DetA	AssA	HOTA	HOTA^{obj}	mAP	DetA	AssA	HOTA	HOTA^{obj}	mAP
Exemplar-guided	Mask	STCN [2]	46.1	54.9	49.8	59.2	33.3	44.7	62.5	52.2	60.1	27.5	46.4	53.1	49.2	54.8	34.8
		Box Tracker* [6]	15.1	22.5	18.0	47.7	12.5	28.7	46.3	35.8	52.3	12.5	11.8	16.6	13.6	18.8	4.4
	Box	STCN (PointRend [7])	41.1	51.1	45.8	57.1	26.7	40.9	60.2	48.9	58.5	23.5	41.1	48.8	44.3	50.9	27.5
		STCN (Matched Det*)	20.0	32.4	24.5	54.4	13.3	38.7	60.8	47.6	57.6	22.7	15.4	25.3	18.7	34.2	11.0
		Box Tracker*	10.9	18.8	13.7	48.6	4.4	27.1	45.0	34.2	52.8	10.8	6.9	12.2	8.6	18.3	2.8
	Point	STCN (Matched Det*)	19.9	32.1	24.4	48.4	12.8	34.4	59.0	44.0	51.6	20.6	16.3	25.4	19.4	31.4	10.8
		Box Tracker*	9.9	17.6	12.7	43.7	3.2	24.2	43.7	31.7	48.0	9.7	6.3	11.2	7.9	17.1	1.6
Class-guided	Common	STCN Tracker*	-	-	-	-	-	51.7	51.9	51.2	66.8	13.6	-	-	-	-	-
		Box Tracker*						51.7	43.0	45.5	62.9	9.9	-	-	-	-	-
	Long-Tail	STCN Tracker†	8.0	13.4	5.5	13.5	0.9	27.0	13.4	17.5	14.4	0.7	3.2	2.4	2.5	8.9	0.6
		Box Tracker†	8.3	29.5	8.2	18.6	1.4	28.0	29.5	27.0	19.8	3.0	3.4	4.3	3.6	11.7	0.9

Table 2. Extended baseline results for the test set with multiple metrics. Evaluation metrics are reported separately for ‘common’, ‘uncommon’ and ‘all’ classes. Object detector training data: *: COCO, †: LVIS.

		Baseline Method	All					Common					Uncommon				
			DetA	AssA	HOTA	HOTA^{obj}	mAP	DetA	AssA	HOTA	HOTA^{obj}	mAP	DetA	AssA	HOTA	HOTA^{obj}	mAP
Exemplar-guided	Mask	STCN [2]	48.8	57.3	52.4	60.7	34.4	45.0	59.2	51.1	61.0	30.0	49.6	57.0	52.7	59.5	35.3
		Box Tracker* [6]	12.1	17.2	14.1	44.4	3.5	21.9	37.3	28.0	48.5	9.2	10.2	13.2	11.4	16.9	2.4
	Box	STCN (Pointrend [7])	41.9	51.7	46.0	58.1	25.6	42.2	58.0	48.9	59.0	26.2	41.9	50.5	45.4	53.6	25.4
		STCN (Matched Det*)	20.6	32.9	25.0	53.9	13.6	33.1	55.5	41.9	56.6	20.5	18.1	28.4	21.7	36.6	12.2
		Box Tracker*	11.6	16.9	13.6	44.2	3.0	21.5	37.2	27.7	48.3	8.3	9.6	12.8	10.8	17.0	1.9
	Point	STCN (Matched Det*)	20.6	32.5	24.9	49.6	12.5	30.8	53.8	39.5	52.3	17.7	18.6	28.3	22.0	34.6	11.5
		Box Tracker*	8.0	14.1	10.1	41.6	2.5	17.8	36.1	24.4	45.5	6.7	6.1	9.7	7.3	16.1	1.7
Class-guided	Common	STCN Tracker*	-	-	-	-	-	37.1	34.9	34.6	58.8	6.8	-	-	-	-	-
		Box Tracker*	-	-	-	-	-	37.1	35.5	34.3	56.9	6.8	-	-	-	-	-
	Long-Tail	STCN Tracker†	5.4	4.6	4.5	13.4	0.3	19.6	16.7	17.1	14.3	1.7	2.6	2.2	2.0	8.0	0.0
		Box Tracker†	5.4	6.8	5.7	15.4	0.3	19.6	23.2	20.1	16.3	1.7	2.6	3.6	2.9	10.7	0.0

Table 3. Extended baseline results for the open-world class-guided task for the validation set.

Baseline Method	All			Common			Uncommon		
	DetRe	AssA	OWTA	DetRe	AssA	OWTA	DetRe	AssA	OWTA
STCN Tracker	67.0	62.6	64.6	78.8	64.1	71.0	20.0	33.3	25.0
Box Tracker	66.9	55.8	60.9	78.7	57.1	60.9	20.1	30.5	24.0
OWTB [9]	70.9	45.2	56.2	76.8	47.0	59.8	46.5	34.3	38.5

Table 4. Extended baseline results for the open-world class-guided task for the test set.

Baseline Method	All			Common			Uncommon		
	DetRe	AssA	OWTA	DetRe	AssA	OWTA	DetRe	AssA	OWTA
STCN Tracker	61.6	54.1	57.5	71.5	55.7	62.9	21.0	28.6	23.9
Box Tracker	61.5	51.1	55.9	71.4	52.5	61.0	21.1	30.0	24.6
OWTB [9]	70.7	45.5	56.3	76.6	47.3	59.9	45.7	33.6	38.3

pothesized earlier).

Finally, we note that some methods have an mAP score of 0.0 for some settings *e.g.* uncommon class set for the long-tail class-guided task (Table 1). This happens when the predicted object tracks cannot be associated with any ground-truth track because their IoUs are all below the acceptance threshold. Note that mAP is computed as an average over several different threshold values, so an overall mAP of 0.0 implies that even the lowest threshold was not satisified by any predicted track in the entire dataset. This further highlights the potential for improvement for the long-tail task, both in terms of per-frame object detections as well as temporal association.

2.2. DetA vs. AssA

The ability to numerically quantify per-frame detection and temporal association quality can prove useful in analyzing strengths and weaknesses of various methods. For instance, when comparing STCN Tracker and Box Tracker for class-guided tasks, we note that the difference in the final HOTA scores arises mainly from the the difference in AssA. This is understandable because both methods use the same set of per-image detection masks. Any small differences in DetA arise from the fact that we perform a post-processing step where very short object tracks (*e.g.* those containing just one detection) are discarded.

2.3. HOTA vs. $\mathrm{HOTA}^{\mathrm{obj}}$

In general, we note that the $\mathrm{HOTA}^{\mathrm{obj}}$ scores are higher than HOTA. We recap that the former gives equal weight to each object track when computing the final score, whereas the latter gives equal weight to each object class, even if the number of ground-truth tracks in the object classes are unbalanced. As a result, the HOTA score is pushed down by the poor performance of the method on a few object classes even though they contain only a few object tracks.

2.4. Comparing Open-world to Other Tasks

In the main text, we discussed how the open-world class-guided task uses OWTA as an evaluation metric, which is a modified version of HOTA where DetA is replaced with DetRe (Detection Recall). In other words, DetRe is a modification of DetA in which false positives are not penalized.

To quantitatively compare the open-world result to other tasks, we can analyze the difference in DetA and DetRe for the same class split and baseline method. We see that the DetRe is consistently higher than DetA: for STCN Tracker on the validation set, DetRe is 78.8 for the 'common' class set (Table 3) compared to a DetA of 51.7 for the common class-guided task (Table 1). The numerical difference between these two arises due to the presence of false positives in the image-level object detector output.

Secondly, we point out that the metric OWTA can be seen as an upper-bound for the $\mathrm{HOTA}^{\mathrm{obj}}$ metric which neglects penalization for false positives. Hence, the 52.3 $\mathrm{HOTA}^{\mathrm{obj}}$ score achieved by STCN tracker for the common class-guided task (Table 1) is lower than the 71.0 OWTA achieved by STCN Tracker on common classes for the open-world task (Table 3). This difference is again attributable to the presence of false positive detections in the method's predicted output.

3. Baseline Runtime

Inference speed for the baselines are given in Table 5

Table 5. Run-time speed for baselines.

Baseline	Image Detector	Runtime (FPS)
STCN [2]	-	3.12
Box Tracker	Mask2Former [1]	1.35
STCN Tracker	Mask2Former	0.95
Box Tracker	Mask-RCNN [5]	3.90
STCN Tracker	Mask-RCNN	1.76

4. Miscellaneous Implementation Details

Point Selection Criterion for Exemplar-guided Task. For the point exemplar-guided task, we select the point coordinate to provide to the method as follows: we compute the distance between all points inside an object mask to the closest point on the object boundary and choose the point with the highest distance, *i.e.* the 'inner-most' point of the object mask. If multiple points share the highest distance value, we select the point closest to the centroid of the object mask.

Image-level Object Detector. As mentioned in the main text, for obtaining image-level object masks for common classes, we employ a Mask2Former [1] network trained on COCO [8]. In particular, we use the best-performing model checkpoint provided by the authors which uses a Swin-L [10] backbone. For the long-tail task, we use the best-performing Mask-RCNN [5] model provided by Detectron2 [12] which is trained on LVIS [4] and has a ResNeXt-101 backbone.

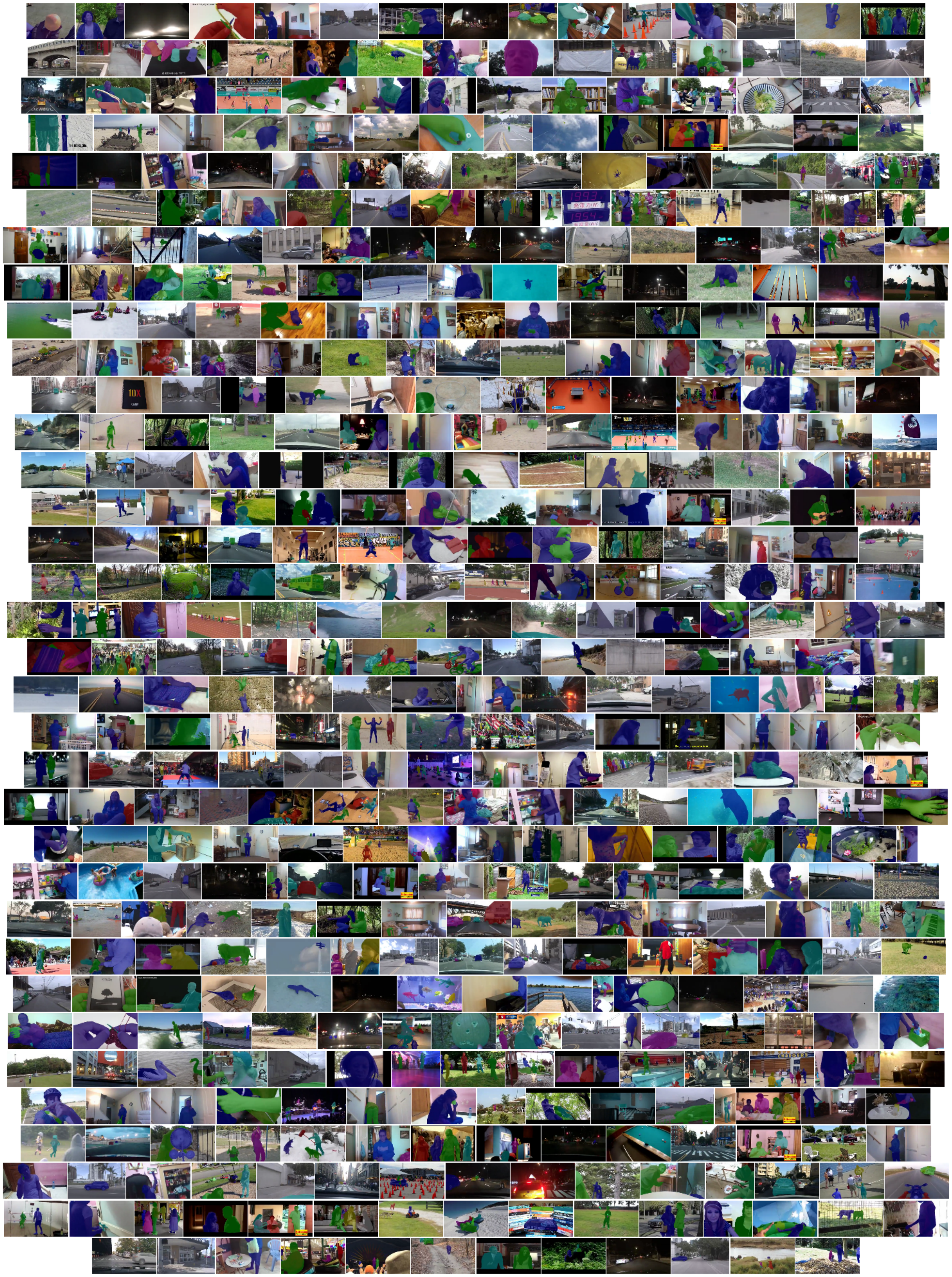

Figure 1. Screenshot of the dataset viewer supplied in the supplemental.zip. Here assorted examples of annotated images from the BURST training set are shown, however in the dataset viewer full videos with tracked mask annotations are shown.

References

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

[2] 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.

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

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

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

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

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

[8] 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.

[9] 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.

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[11] 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.

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

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