TAO: A Large-Scale Benchmark for Tracking Any Object Supplementary Material

Achal Dave[1], Tarasha Khurana[1], Pavel Tokmakov[1]
Cordelia Schmid[2], and Deva Ramanan[1,3]

[1] Carnegie Mellon University
[2] Inria [3] Argo AI

Section 1 further analyzes TAO annotations, including quality control and statistics. Section 2 further analyzes metrics, comparing 3D IoU to MOT challenge [19] metrics. Finally, Section 3 further analyzes tracking methods, providing results on non-LVIS categories, improved initialization for user-initialized trackers, and hyperparameter tuning experiments.

1 TAO annotations

This section presents additional details about TAO annotations. Section 1.1 assesses the diversity and quality of annotations. Section 1.2 analyzes the size, length and motion statistics of labeled tracks. Finally, Section 1.3 provides further information regarding the construction of dataset splits.

1.1 Annotation diversity and quality

We analyze the diversity and quality of TAO annotations by re-annotating 50 videos in the dataset.

Diversity. One might hope that this re-annotation closely matches the original annotation. However, in our federated setup, annotators are instructed to label only a subset of moving objects in each video. Thus, the annotations would only match if annotators had a bias towards a specific set of objects, which would hurt the diversity of TAO annotations. To verify whether this is the case, we check whether each track in the re-annotation corresponds to an object labeled in the original annotation. Concretely, if a re-annotated track has high overlap (IoU > 0.75) with a track in the original annotation, we assume the annotator is labeling the same object. Our re-annotation results in 310 tracks from 50 videos. Of these 310 tracks, just over half (177, or 57%) overlapped with those in the initial labeling with IoU > 0.75. The rest were *new* objects not originally labeled in TAO, suggesting that annotators chose to label a diverse selection of objects.

Quality. Next, we evaluate the annotation agreement of the 177 re-annotated tracks that correspond to tracks originally labeled in TAO. If our annotations are of high quality, we expect these tracks to have a very high IoU (say, > 0.9), as well as matching class labels. Indeed, the average IoU for the 177 overlapping

tracks was 0.93, indicating annotators precisely labeled the spatial and temporal extent of objects. Finally, we evaluate the quality of the class labels in TAO. 165 (93%) were labeled with the same category as in the initial labeling; an additional 6 (3%) were labeled with a more precise or more general category (e.g., 'jeep' vs. 'car'); finally, 6 were labeled with similar labels (e.g., 'kayak' vs. 'canoe') or other erroneous labels. This analysis indicates that despite the large vocabulary in TAO, the class labels in TAO are of high quality.

If our annotations are of high quality, we expect these tracks to have a very high IoU (say, > 0.9), as well as matching class labels.

Annotation details. We worked closely with a professional data-labeling company, Scale.ai, to label TAO. Each track was labeled by a Scale annotator, reviewed by Scale reviewers, and finally manually inspected by the authors.

1.2 Annotation statistics

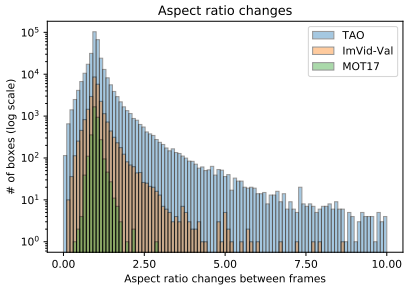
We present further analysis of the annotated tracks in TAO in Figure 1. We compare TAO to MOT-17 [19] and ImageNet-Vid [22], which are benchmark datasets where the Viterbi [10,8] and the Tracktor [1] approaches were originally evaluated.

Figure 1(a) shows the distribution of changes in aspect ratio between two annotated frames at 1FPS. Concretely, the aspect ratio change is $(w_t/h_t)/(w_{t-1}/h_{t-1})$, where w_t, h_t are the width and height of the object at time t, respectively (see [16]). This metric can be used to understand the types of motion in tracking datasets. MOT-17 focuses on people, which largely have the same aspect ratio over time. ImageNet-Vid has a slightly more diverse distribution of changes in aspect ratio, but TAO has by far the most diverse distribution, due to its large size and diversity of categories.

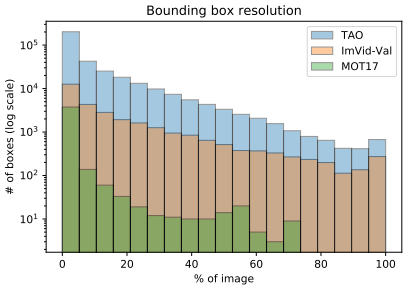
Figure 1(b) plots the distribution of bounding box resolution as a percentage of the image. MOT-17 tends to have smaller bounding boxes, while TAO and ImageNet-Vid have a variety of object sizes. Note again that TAO presents a much larger number of tracks used for evaluation, visible even on the log-scale in Figure 1(b), than ImageNet-Vid val.

Figure 1(c) presents the distribution of object motion, proportional to the size of the object. Concretely, let a_t be the area of the bounding box at time t. We define the distance in x as $d_t^x = \frac{\|x_t - x_{t-1}\|}{a_{t-1}}$, and similarly for d_t^y. Then, $d_t = \|[d_t^x, d_t^y]\|_2^2$. As with Figure 1(a), we plot these changes at 1FPS so that the annotation rate does not impact the plot. We note that TAO contains a variety of object motions, including extremely fast motions for small objects, as evidenced by the number of boxes with motion change larger than 5.0.

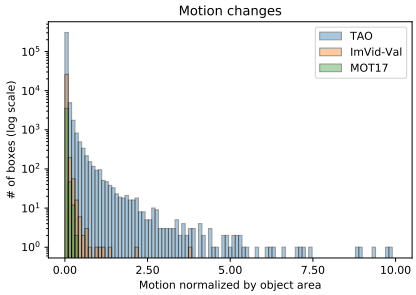
Figure 1(d) shows the distribution of object track lengths in TAO. For clarity, we group the tracks into 3 bins based on length: short, medium and long, which correspond to less than 1/3, between 1/3 and 2/3, and greater than 2/3 of the length of the video. The plot shows that TAO provides diversity in object track length, requiring methods to be able to track for long periods of time, while also



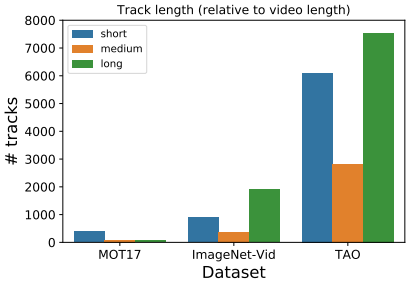
(a) Ratio between aspect ratio of bounding boxes between two consecutive annotated frames at 1FPS.



(b) Bounding box size counts relative to size of the image.



(c) Distance between center of objects between two consecutive annotated frames at 1FPS.



(d) Track length counts, relative to video lengths.

Fig. 1. Additional statistics of the TAO dataset. See Section 1.2 for details.

being able to recognize when an object is missing. By contrast, MOT-17 is biased towards short tracks, while ImageNet-Vid is biased towards long tracks.

Finally, we present statistics of recent benchmarks for user-initialized tracking (or single-object tracking) in Table 1. We note that datasets tend to benchmark tracking on a smaller number of categories than TAO, and on far fewer videos. While this may be appealing from a computational perspective, we argue that progress in tracking requires evaluating on a large, diverse set of scenarios, ensuring that methods do not overfit to any small set of videos or environments. Further, unlike standard user-initialized tracking datasets, TAO contains nearly 5x as many tracks per video, leading to a much larger number of total tracks compared to prior benchmarks.

1.3 Split construction

We construct our 'train', 'val', and 'test' splits to respect the following constraints:

Table 1. Statistics of major user-initialized tracking datasets.

Dataset	Classes Eval.	Classes Train	Videos Eval.	Videos Train	Avg length (s)	Tracks / video	Min resolution	Ann. fps	Total Eval length (s)
VOT 2019 LT [15]	0	16	0	50	143.5	1	290x217	~30	7,176
GOT-10k [14][a]	84	480	360	9,335	12.2	1	270x480	10	4,384
OxUvA [23]	22	0	366	0	141.2	1.1	192x144	1	51,667
LaSOT [7]	70	70	280	1,120	82.1	1	202x360	~25	23,520
TrackingNet [20]	27	27	511	30,132	14.7	1	270x360	~28	7,511
TAO (Ours)[b]	785	316	2,407	500	36.8	5.9	640x480	1	88,605

[a] Stats from the GOT-10k dataset release, which differ from those in [14].
[b] TAO train and eval contain partially overlapping subsets of the overall 833 categories.

- **Charades** contains videos recorded by mechanical turk workers, and one worker may contribute multiple videos to Charades. We ensure that any two videos uploaded by the same worker falls in the same split.
- **ArgoVerse** contains video recordings from different cameras from the same driving sequence. We ensure that all videos from the same driving sequence fall in the same split.
- **HACS** contains videos uploaded to YouTube. Any two videos uploaded by the same YouTube user, or uploaded to the same YouTube channel, must fall in the same split.
- **AVA.** We split AVA movies into multiple contiguous shots, and ensure shots from the same movie fall in the same split.
- **YFCC100M** contains videos uploaded to Flickr. Any two videos uploaded by the same Flickr user fall in the same split.
- **BDD** and **LaSOT**: No constraints are applied for split construction.

2 Metrics

In this section, we further analyze the 3D IoU metric (2.1), report results using the MOT challenge [19] metrics (2.2), and finally present per-category APs for SORT (2.3).

2.1 3D IoU Discussion

The mAP metric using 3D IoU provides a concise, interpretable evaluation of tracking in the wild, as evidenced by its use in recent datasets for multi-object tracking with many categories [6,27]. We further discuss this metric below:

Relation to identity swaps. Figure 2 shows that 3D IoU is correlated with a key metric for tracking: identity swaps, as measured by the MOT challenge [19] metrics.

Partial credit. Evaluating trackers with mAP requires specifying an IoU threshold, which we set to 0.5 throughout the experiments in the main paper. Consequentially, trackers do not receive *partial credit* for tracking an object for short

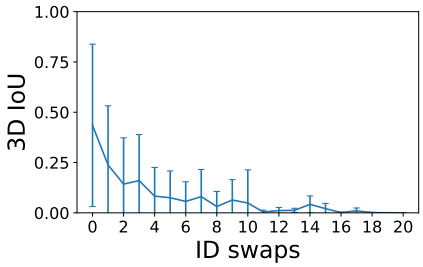


Fig. 2. For each pair of predicted and groundtruth tracks matched to each other on TAO, we compute the 3D IoU and number of ID swaps. Above, we plot the mean and variance of 3D IoU vs. ID swaps across tracks, and show that 3D IoU drops as the number of ID swaps increases.

time periods. Consider two trackers: Tracker A perfectly tracks an object for 30% of its track length, while Tracker B only tracks the object 5% of the time. At an IoU threshold of 0.5, A and B will result in the same mAP. By contrast, metrics such as MOTA and ID-F1 will be significantly higher for A than for B. The 3D IoU mAP metric takes inspiration from image-based detection metrics: as object detectors receive no credit for loose localizations, object trackers receive no credit for loosely tracking objects for a few frames. If desired, the mAP metric can be modified to provide partial credit by averaging over multiple IoU thresholds, similar to the COCO evaluation [18].

Confidence estimates. Metrics such as MOTA [2] and ID-F1 [25] metrics do not evaluate the confidence provided by many modern tracking approaches. By contrast, our mAP metric evaluates these explicitly when tracing out the precision-recall curve. This allows us to evaluate methods across diverse application scenarios, which may have different tradeoffs between precision and recall.

Impact of object size. 3D IoU is computed over spatio-temporal volumes. As such, frames where an object's bounding box is *large* have a greater impact on the spatio-temporal volume than frames where an object's bounding box is small, thus factoring in more heavily into the IoU measure. We note that for many applications, such as navigation, this is a desirable property, as accurate localization and tracking is more important for nearby objects. For other applications, additional diagnostics, such as MOTA (Section 2.2), can be used for further analysis.

2.2 MOTA results

For completeness, we present results using the MOT challenge suite of metrics [19]: MOTA [2], ID-F1 [21], mostly-tracked (MT) tracks and mostly-lost (ML) tracks [25], false-positives (FP), false-negatives (FN) and identity swaps (ID Sw.), computed using the `py-motmetrics` library [13]. To do this, we first make two modifications to the MOT metrics:

Table 2. Results from tuning track score thresholds for multi-object trackers, user-initialized trackers, and Tracktor++ on TAO train, reporting MOTA.

Tracker	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Detector	-18.1	-9.7	-6.1	-3.8	-2.2	-1.3	-1.0	-0.3	-0.01	**0.0**
SORT	-3.0	7.7	7.7	7.9	**8.5**	6.9	5.4	3.6	2.5	0.0
Viterbi	-8.4	2.5	5.4	5.6	6.2	**6.8**	5.3	5.3	3.3	0.0
ATOM	21.8	21.8	21.8	21.8	21.8	21.8	**27.2**	19.8	8.2	0.0
DIMP	22.7	22.7	22.7	22.7	22.7	**22.7**	22.6	21.4	20.3	19.1
ECO	0.7	0.7	0.3	1.5	7.0	8.1	**12.6**	6.1	0.3	0.0
SiamMask	19.7	19.7	19.7	19.7	19.7	19.7	19.7	19.7	**19.9**	0.0
SiamRPN++	21.0	21.0	21.0	21.0	21.0	21.0	21.0	21.0	**25.5**	0.0
SiamRPN++ LT	22.9	22.9	22.9	22.9	22.9	22.9	22.9	22.9	**22.9**	0.0
Person-only evaluation										
Tracktor++	65.9	66.0	66.2	66.5	67.2	67.9	**68.4**	67.8	63.0	

Federated MOTA and ID-F1. We update the MOTA and ID-F1 metrics for a federated dataset by only counting false positives (FPs) for a category c in video v if we know that all instances of category c are annotated in video v (i.e., if v is in P_c or N_c as defined in Sec. 3 of our paper). While this approach is not perfect, as it can over-estimate the performance of a tracker, it provides a simple adaptation to the federated setup.

Multiple categories. The MOT metrics are usually reported for a single category [19], or separately for a small number categories [9]. This is not a scalable strategy for TAO, which contains 833 categories. Instead, we compute metrics separately per category, and combine them across categories. Concretely, for metrics such as MOTA and ID-F1, we report the average value across categories. For counters, including MT (mostly-tracked), ML (mostly-lost), FP (false-positives), FN (false-negatives) and ID Sw. (identity switches), we report the sum across categories. Note that while MOTA and ID-F1 are balanced across categories, the 'counters' are heavily dominated by the most frequent categories.

Thresholds. Unlike mAP, the MOT metrics require picking a confidence threshold for evaluation. To do this, we search over track score thresholds on TAO train and report results in Table 2. For Viterbi and user-initialized trackers, the track score threshold is applied after the tracker per-frame score threshold tuned in Section 3.3. Hence, the MOTA for track thresholds below the per-frame threshold are equivalent (e.g., for DIMP, the optimal per-frame threshold is 0.5, and so the MOTA for thresholds below 0.5 is exactly the same: 22.7).

We use the optimal thresholds from the train set to report results on the validation set for multi-object trackers in Table 3, for user-initialized trackers in Table 4, and for person-tracking in Table 5. In general, we find that the conclusions drawn in our main paper using mAP are consistent with experiments using MOTA, with two exceptions.

Table 3. MOT challenge metrics for multi-object trackers on TAO validation. As the 'Track' oracle implicitly removes false positive detections, we set score thresholds to 0 when it is used.

Method	Oracle Class	Oracle Track	MOTA ↑	ID-F1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
Detection			-2.3	1.3	1,495	1,941	3,492	60,776	48,377
Viterbi			5.6	10.0	1,407	2,409	5,367	62,341	10,262
SORT			6.7	10.4	1,687	2,117	4,146	59,481	4,772
Detection		✓	38.8	48.4	2,191	919	0	42,796	0
Viterbi	✓		8.3	13.8	1447	2361	5595	60787	10292
SORT	✓		11.3	15.6	1,725	2,066	4,165	58,418	4,773
Detection	✓	✓	83.2	89.6	3,806	188	0	17018	6

User init. First, Table 4 shows that user-initialized trackers provide significant improvements over SORT using MOTA and ID-F1, while this did not hold for mAP. These metrics provide partial credit for tracking objects for short periods of time, while mAP (with an 3D IoU threshold of 0.5) requires tracking an object for at least half its track length (see Section 2.1). One can obtain mAP rankings consistent with MOTA/ID-F1 by using an artificially low IoU threshold; at a threshold of 0.1, DIMP strongly outperforms SORT, 71.0 mAP to 36.9 mAP. These results reinforce the notion that user-initialization is helpful for tracking short periods after initialization, but less helpful in the long term.

Table 4. MOT challenge metrics on TAO validation, comparing user-initialized trackers with SORT using a class oracle.

Method	Oracle Box Init	Oracle Class	MOTA ↑	ID-F1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
SORT		✓	11.3	15.6	1,725	2,066	4,165	58,418	4,773
ECO	✓	✓	11.8	24.0	753	4341	5395	85415	42
SiamRPN++ LT	✓	✓	13.1	54.0	2,292	753	19282	42255	2103
SiamRPN++	✓	✓	14.6	49.9	2,110	1229	16630	45612	1411
ATOM	✓	✓	16.9	46.7	1,694	2,274	14,625	55,875	481
DIMP	✓	✓	24.4	55.1	2,279	870	16,966	42,729	1,290

MOTA-Person. Second, as noted in the main paper, Table 5 shows that MOTA-person is significantly higher than MOTA-overall (6.7 vs 54.8 for SORT), whereas the delta is smaller under mAP (13.2 vs 18.5 for SORT). We find MOT metrics heavily reward accurate detection while 3D IoU heavily penalizes inaccurate tracking. Because person detectors strongly outperform other category detectors on average, this is manifested as a high MOTA-person score.

Table 5. MOT challenge metrics on TAO validation for the 'person' category.

Method	MOTA ↑	ID-F1 ↑	MT ↑	ML ↓	FP ↓	FN ↓	ID Sw. ↓
Viterbi	44.5	50.4	939	741	21,678	3,167	7,128
SORT	54.8	56.2	1,078	542	20,025	2,432	3,567
Tracktor++	66.6	64.8	1,529	411	12,910	2,821	3,487

Other benchmarks. Finally, we directly compare Tracktor++ on TAO with its performance on the MOT-17 dataset. Table 6 shows that the more sophisticated components of Tracktor++ (re-identification and motion compensation) lead to significant improvements on TAO, suggesting TAO encourages trackers robust to common tracking challenges, including occlusion and camera motion.

Table 6. MOTA on TAO val vs. MOT-17, for Tracktor. TAO encourages trackers robust to camera motion and occlusion, as noted by the significant improvement to Tracktor using the reID and camera motion compensation (CMC) components.

	TAO		MOT-17	
Method	train	val	train	test
Tracktor	63.8	61.6	61.5	-
Tracktor++ (reID + CMC)	68.4	66.6	61.9	53.5

2.3 AP per category

We present per-category APs in Figure 3 for the SORT algorithm reported in the main paper, though we note that AP for individual categories can be noisy in a federated setup [11]. Note that for 180 categories, this algorithm achieves 0 AP; for conciseness, we plot only the categories with non-zero AP.

3 Additional tracking results

Section 3.1 presents results for user-initialized trackers on *all* categories in TAO, Section 3.2 analyzes the improvement to user-initialized trackers by using a more informative initialization. Section 3.3 reports results from tuning trackers on TAO train.

3.1 User-initialized trackers on all categories

In the main paper, we focus our analysis on a subset of TAO categories which exist in the LVIS [11] dataset, allowing us to repurpose existing object detectors for multi-object tracking. Here, we evaluate user-initialized trackers (which do

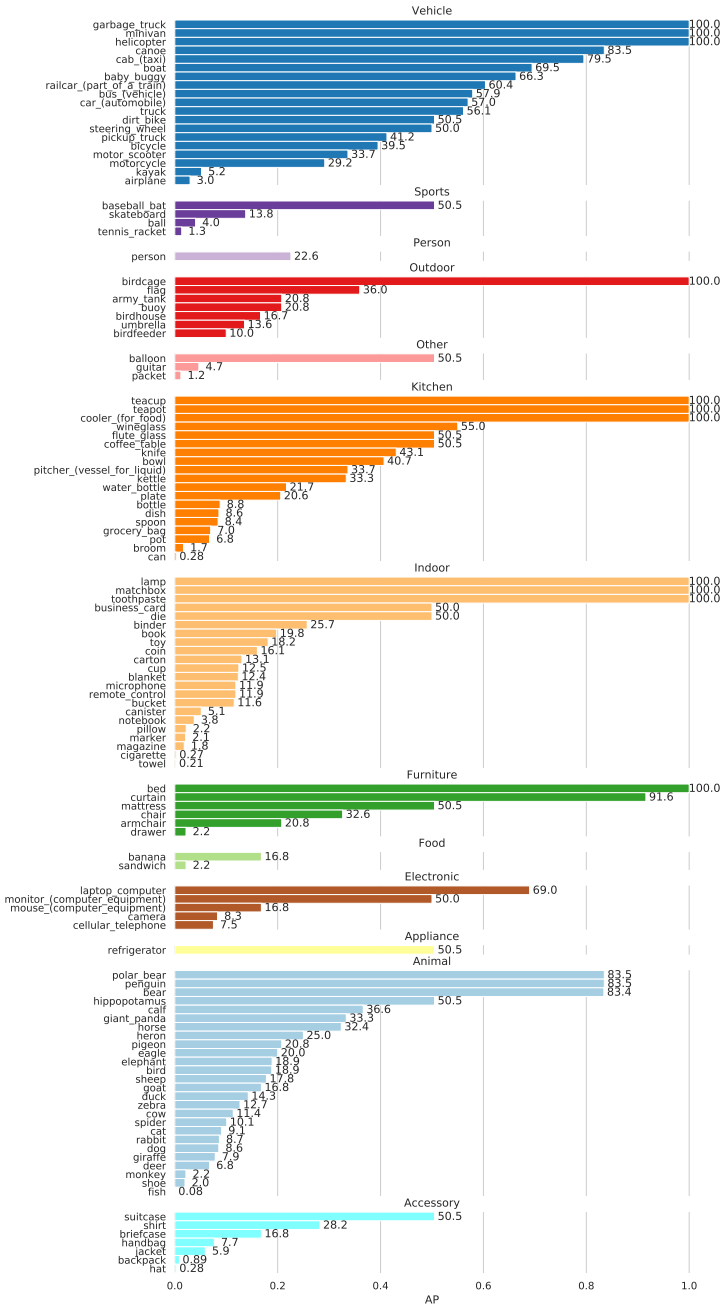


Fig. 3. Per-category AP for the SORT algorithm, omitting 180 categories which result in zero AP for conciseness. As common in large-vocabulary datasets (LVIS, ADE-20K, LabelMe), average accuracy is dominated by classes in the tail, many of which result in 0 AP. Note that AP for individual categories can be noisy in a federated setup [11].

not require object detectors) on the remaining categories in TAO (Table 7). We generally find that the results are consistent with the results on the LVIS categories.

Table 7. Results on non-LVIS, free-form text categories in TAO validation.

Method	Non-LVIS categories, validation
ECO	24.1
SiamMask	27.0
SiamRPN++	27.7
SiamRPN++ LT	25.1
ATOM	29.5
DIMP	**29.6**

3.2 Improved initialization for user-initialized trackers

The standard approach for initializing user-initialized trackers (denoted 'Init first') initializes trackers using the first frame an object appears in, and runs trackers for the rest of the video. As the object may be partially occluded in this first frame, we additionally report a variant in Table 8 which initializes trackers using the frame with the largest bounding box (denoted 'Init biggest'), and runs trackers forwards and backwards in time. The 'Init biggest' strategy provides stronger improvements over SORT by initializing with easier frames, but cannot be used in *online* applications, as it requires access to the entire video.

3.3 Hyperparameter tuning

This section reports detailed results of tuning each tracker on TAO train, as well as information about the detector used for SORT, Viterbi and Tracktor++ (3.4).

Preliminary: Score thresholds. Before discussing the details of each tracker, we define three different score thresholds used by trackers, and refer to them by name throughout the appendix:

1. Detection score: This is the confidence reported by a detector for each object at each frame, *before* any tracking has taken place.
2. Tracker per-frame score: This is the confidence reported by the tracker for each object at each frame, *after* tracking is complete.
3. Track score: This is the confidence reported by the tracker for each object *track* throughout the video. This confidence is used to rank tracks when computing mAP. When computing MOTA, we tune the threshold for reporting tracks using the track score, as described in Section 2.2.

SORT. We tune three parameters internal to SORT, as well as parameters of the underlying detector in Table 9. We tune the following SORT parameters:

Table 8. User-initialized tracking results on 'val', reporting with a more informative initialization strategy ('Init biggest', see **??**), which provides improvements for user-initialized trackers. Because some user-initialized trackers are trained on videos in TAO, we re-train them on their original train set with TAO videos removed, denoting this with *.

Method	Oracle		Track mAP	
	Box Init	Class	Init first	Init biggest
SORT		✓	30.2	
ECO [5]	✓	✓	23.7	30.4
SiamMask [24]	✓	✓	30.8	37.0
SiamRPN++ LT [17]	✓	✓	27.2	30.4
SiamRPN++ [17]	✓	✓	29.7	35.9
ATOM* [4]	✓	✓	30.9	**38.6**
DIMP* [3]	✓	✓	**33.2**	38.5

1. Det / image: Max number of detections output by the detector per image.
2. Detection score
3. `max_age`: How many frames tracks are kept 'alive' for, without any detections being matched to them.
4. `min_hits`: How many frames a track must be alive for before it is considered 'confirmed' and output.
5. `min_iou`: Minimum IoU between a track and a detection required for linking the two.
6. NMS Thresh: The NMS IoU threshold used by the detector. We experiment with more aggressive NMS, which may make the task of linking detections using IoU easier.

The first row in Table 9 corresponds to the default SORT parameters. Due to the significant motion and long duration of sequences in TAO (see Section 1.2), we find that increasing `max_age` and decreasing `min_iou` and `min_hits` helps significantly with accuracy. Additionally, we find that outputting more boxes per image consistently improves accuracy. Lowering the score threshold from 0.1 to 0.0005 results in a 2.1 point improvement from 8.2 to 11.3, and lowering the NMS and score thresholds provides even more significant improvements, from 11.3 to 16.3.

Viterbi. The Viterbi approach has a number of tunable parameters. Unfortunately, the code for this approach is prohibitively expensive to run, taking over a week of compute time to process TAO train in parallel on 4 machines. Due to this constraint, we do not tune the internal parameters of this approach. However, Table 10 shows that tuning the tracker's per-frame score post-hoc can provide small improvements in accuracy, from 8.5 to 9.0.

Tracktor++. Tracktor++ by default thresholds the output of a detector at 0.5. Table 11 shows the results of tuning this threshold on TAO train. Perhaps surprisingly, we find that Tracktor++ is fairly robust to this parameter, unlike

Table 9. Results from tuning SORT parameters (by coordinate descent) on TAO train, where the active coordinate (parameter) is highlighted.

Params						
NMS Thresh	Det / image	Det score	`max_age`	`min_hits`	`min_iou`	Track mAP
0.5	300	0.1	1	3	0.3	4.3
0.5	300	0.1	1	3	0.1	5.0
0.5	300	0.1	1	3	0.5	4.3
0.5	300	0.1	1	1	0.1	5.1
0.5	300	0.1	1	5	0.1	4.9
0.5	300	0.1	1	10	0.1	4.9
0.5	300	0.1	10	1	0.1	6.5
0.5	300	0.1	50	1	0.1	8.1
0.5	300	0.1	100	1	0.1	8.2
0.5	300	0.001	100	1	0.1	10.5
0.5	300	0.0005	100	1	0.1	11.3
0.5	300	0.0001	100	1	0.1	10.9
0.5	10,000	0.0005	100	1	0.1	9.4
0.1	10,000	0.0005	100	1	0.1	15.3
0	**10,000**	**0.0005**	**100**	**1**	**0.1**	**16.3**

Table 10. Results from tuning the Viterbi tracker's per-frame score threshold TAO train.

Tracker per-frame score	0	**0.1**	0.2	0.3	0.4	0.5
Track mAP	8.5	**9.0**	8.4	8.4	7.8	7.3

SORT (as seen in Table 9). We hypothesize that this may be because of two Tracktor++ components: (1) the use of detections at time t as proposal at time $t+1$ may make detectors more likely to consistently output high-confidence detections for tracks, and (2) the re-id component may allow Tracktor++ to more accurately recover tracks with no matching detections for a few frames.

Table 11. Results from tuning Tracktor's detection score threshold on TAO's train set.

Detection score	0.1	0.2	0.3	**0.4**	0.5	0.6	0.7	0.8	0.9
Track mAP	35.1	35.5	35.5	**35.7**	35.0	34.7	34.6	33.0	29.8

User-initialized trackers. As user-initialized trackers do not explicitly report when an object is *absent*, we modify each method to report an object as absent when the confidence drops below a threshold. We tune this threshold on TAO

Table 12. Results from tuning user-initialized trackers' per-frame score threshold on TAO train.

Tracker	Tracker per-frame score										
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.99
ATOM	34.3	34.3	36.2	36.6	**36.7**	34.4	31.9	26.3	17.8	9.9	2.1
DIMP	31.2	33.2	35.3	36.1	36.2	**36.4**	34.8	33.2	30.3	27.8	19.7
ECO	25.4	25.4	26.3	**27.3**	27.1	25.6	20.6	14.7	8.9	3.0	2.7
SiamMask	27.9	28.6	28.8	28.8	29.3	29.4	30.0	30.7	**30.9**	30.5	27.3
SiamRPN++	28.6	29.2	29.3	30.3	30.0	30.9	31.1	**31.5**	31.2	31.4	28.1
SiamRPN++-LT	27.0	26.6	27.1	27.2	27.0	27.7	28.0	27.9	28.0	**28.2**	26.7

train. Table 12 shows that the optimal threshold varies by tracker, and tuning this parameter can lead to significant changes in accuracy (e.g., 5.2% in the case of DIMP when using a threshold of 0.5 as opposed to the default of 0).

Fig. 4. Qualitative comparison between a Mask R-CNN model trained on LVIS (left) and one trained on LVIS+COCO (right). Training on additional COCO data is critical for accurately detecting common categories, such as people and cars.

3.4 Detector details

Throughout our experiments, we used a Mask R-CNN model [12] using a ResNet-101 backbone. We re-train this model on a combination of the LVIS and COCO datasets (described below) using the default training parameters for training on LVIS (including repeat factor sampling). Specifically, we used the detectron2 repository [26], with the configuration file at https://github.com/facebookresearch/detectron2/blob/b6fe828a2f3b2133f24cb93c1d0d74cb59c6a15d/configs/LVIS-InstanceSegmentation/mask_rcnn_R_101_FPN_1x.yaml.

We found that training on a combination of COCO and LVIS annotations leads to a noticeable improvement in detection quality, which is particularly significant

for people, compared to training on LVIS alone. To build this combination, we add COCO annotations to every image in the LVIS dataset. To avoid duplicates, we remove COCO annotations that have IoU > 0.7 with an LVIS annotation. We show qualitative results of this improvement in Figure 4.

References

1. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: ICCV (2019) 2
2. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: the clear mot metrics. Journal on Image and Video Processing **2008**, 1 (2008) 5
3. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: CVPR (2019) 11
4. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: CVPR (2019) 11
5. Danelljan, M., Bhat, G., Shahbaz Khan, F., Felsberg, M.: Eco: Efficient convolution operators for tracking. In: CVPR (2017) 11
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 4
7. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: LaSOT: A high-quality benchmark for large-scale single object tracking. In: CVPR (2019) 4
8. Feichtenhofer, C., Pinz, A., Zisserman, A.: Detect to track and track to detect. In: ICCV (2017) 2
9. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 6
10. Gkioxari, G., Malik, J.: Finding action tubes. In: CVPR (2015) 2
11. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019) 8, 9
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) 13
13. Heindl, C., Valmadre, J.: py-motmetrics. https://github.com/cheind/py-motmetrics/ (2019) 5
14. Huang, L., Zhao, X., Huang, K.: GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. arXiv preprint arXiv:1810.11981 (2018) 4
15. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Čehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., Eldesokey, A., Kapyla, J., Fernandez, G.: The seventh visual object tracking vot2019 challenge results (2019) 4
16. Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. TPAMI **38**(11), 2137–2155 (2016) 2
17. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR (2019) 11
18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014) 5
19. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016) 1, 2, 4, 5, 6

20. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV (2018) 4
21. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV (2016) 5
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015) 2
23. Valmadre, J., Bertinetto, L., Henriques, J.F., Tao, R., Vedaldi, A., Smeulders, A.W., Torr, P.H., Gavves, E.: Long-term tracking in the wild: A benchmark. In: ECCV (2018) 4
24. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: CVPR (2019) 11
25. Wu, B., Nevatia, R.: Tracking of multiple, partially occluded humans based on static body part detection. In: CVPR (2006) 5
26. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019) 13
27. Yang, L., Fan, Y., Xu, N.: Video instance segmentation. In: ICCV (2019) 4