International Journal of Computer Vision
https://doi.org/10.1007/s11263-020-01387-y

LaSOT: A High-quality Large-scale Single Object Tracking Benchmark

Heng Fan[1] · Hexin Bai[2] · Liting Lin[3,4] · Fan Yang[2] · Peng Chu[2] · Ge Deng[2] · Sijia Yu[2] · Harshit[1] · Mingzhen Huang[1] · Juehuan Liu[2] · Yong Xu[3,4] · Chunyuan Liao[5] · Lin Yuan[6] · Haibin Ling[1]

Received: 30 April 2020 / Accepted: 18 September 2020

Abstract
Despite great recent advances in visual tracking, its further development, including both algorithm design and evaluation, is limited due to lack of dedicated large-scale benchmarks. To address this problem, we present **LaSOT**, a high-quality **La**rge-scale **S**ingle **O**bject **T**racking benchmark. LaSOT contains a diverse selection of 85 object classes, and offers 1550 totaling more than 3.87 million frames. Each video frame is carefully and manually annotated with a bounding box. This makes LaSOT, to our knowledge, the largest densely annotated tracking benchmark. Our goal in releasing LaSOT is to provide a dedicated high quality platform for both training and evaluation of trackers. The average video length of LaSOT is around 2500 frames, where each video contains various challenge factors that exist in real world video footage,such as the targets disappearing and re-appearing. These longer video lengths allow for the assessment of long-term trackers. To take advantage of the close connection between visual appearance and natural language, we provide language specification for each video in LaSOT. We believe such additions will allow for future research to use linguistic features to improve tracking. Two protocols, *full-overlap* and *one-shot*, are designated for flexible assessment of trackers. We extensively evaluate 48 baseline trackers on LaSOT with in-depth analysis, and results reveal that there still exists significant room for improvement. The complete benchmark, tracking results as well as analysis are available at http://vision.cs.stonybrook.edu/~lasot/.

Keywords Visual tracking · Large-scale benchmark · High-quality dense annotation · Tracking evaluation

1 Introduction

Visual object tracking plays a crucial role in computer vision and has a wide range of applications including intelligent vehicles, robotics, human-machine interaction, and surveillance (Li et al. 2013; Smeulders et al. 2014; Yilmaz et al. 2006). Among various types of tracking problems, a popular and fundamental one is the so-called model-free generic object tracking, which is the focus of this paper. Briefly speaking, given the target bounding box in the initial frame, the goal of tracking is to locate the target in a video sequentially.

In recent years, considerable progress has been made in improving tracking performance. Visual tracking benchmarks have been playing a key role in providing fair comparison and evaluation of different trackers, advancing the research frontier of visual tracking significantly. However, current benchmarks have limited further development of tracking in the deep learning era, as well as more authentic performance evaluation in real world scenarios, due to the following reasons:

Small-scale Motivated by the success of deep learning (Krizhevsky et al. 2012; He et al. 2016; Simonyan and Zisserman 2015), deep feature representation has been widely adopted for target appearance modeling in tracking and has achieved significant improvements. To learn a robust deep representation, a dedicated *large-scale* tracking bench-

Communicated by Konrad Schindler.

H. Fan and H. Bai make equal contribution to this work.

✉ Heng Fan
hefan@cs.stonybrook.edu

Haibin Ling
hling@cs.stonybrook.edu

1 Stony Brook University, Stony Brook, USA
2 Temple University, Philadelphia, USA
3 South China University of Technology, Guangzhou, China
4 Peng Cheng Laboratory, Shenzhen, China
5 HiScene Information Technologies, Shanghai, China
6 Amazon Web Services, Palo Alto, USA

Published online: 29 September 2020

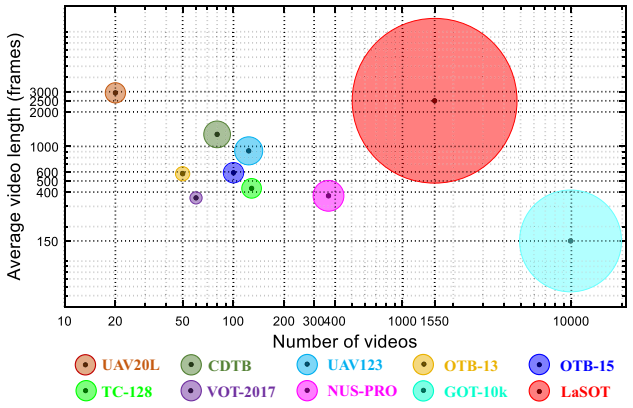


Fig. 1 Summary of existing benchmarks with precise dense (per frame) annotations using log-log scale, containing OTB-13 (Wu et al. 2013), OTB-15 (Wu et al. 2015), TC-128 (Liang et al. 2015), NUS-PRO (Li et al. 2016), UAV123 (Mueller et al. 2016), UAV20L (Mueller et al. 2016), CDTB (Lukezic et al. 2019), VOT-2017 (Kristan et al. 2017), GOT-10k (Huang et al. 2019) and LaSOT. The circle diameter is in proportion to the number of frames of a benchmark. The proposed LaSOT is *larger* than all other benchmarks with more than 3.87 M frames, and focused on *long-term* tracking with average video length of around 2500 frames. Best viewed in color

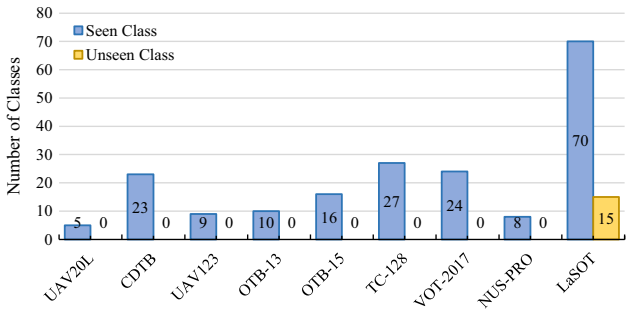


Fig. 2 Comparison of number of classes for evaluation in densely annotated benchmarks, containing UAV20L (Mueller et al. 2016), CDTB (Lukezic et al. 2019), UAV123 (Mueller et al. 2016), OTB-13 (Wu et al. 2013), OTB-15 (Wu et al. 2015), TC-128 (Liang et al. 2015), VOT-2017 (Kristan et al. 2017), NUS-PRO (Li et al. 2016) and LaSOT. We observe that the proposed LaSOT contains the most seen object categories, containing 70 different ones. Moreover, 15 extra unseen object classes are provided for new one-shot evaluation. Note that, GOT-10k (Huang et al. 2019) is not included for comparison because its method to count object classes is different from existing benchmarks

mark is needed. However, most existing datasets contain less than 400 videos (see Figure 1), which makes it hard to learn a *tracking-specific* deep representation. Consequently, researchers in the tracking community have been forced to leverage either the pre-trained models (*e.g.*, (Krizhevsky et al. 2012), (Simonyan and Zisserman 2015) and He et al. (2016)) from ImageNet (Deng et al. 2009) for deep feature extraction or the sequences from video object detection (*e.g.*, Russakovsky et al. (2015) and Real et al. (2017)) for deep feature learning, which may result in suboptimal tracking performance owing to intrinsic differences between different tasks (Yosinski et al. 2014). Extensive evaluation on *large-scale* benchmark is needed to reliably demonstrate performance and generality of trackers.

Lack of high-quality dense annotations Accurate and dense (*i.e.*, per-frame) annotations are crucial to visual object tracking for several reasons. They ensure more accurate and reliable evaluations and more fair comparisons for different trackers, offer desired training samples for developing tracking algorithms, and provide rich motion information and temporal context in videos. It is worth noting that there have been benchmarks proposed recently built towards large-scale and long-term tracking, such as Müller et al. (2018) and Valmadre et al. (2018). However, their annotations are either semi-automatic (*e.g.*, generated by a tracking algorithm) or sparse (*e.g.*, labeled every 30 frames), limiting their usability.

Short-term tracking In order to be deployed in practical application, a tracking algorithm should be able to work well in a long sequence where the target object may frequently leave and enter the view. However, most current tracking datasets contain shorter length videos making them *short-term* benchmarks. As shown in Fig. 1, the average video length of these benchmarks is less then 600 frames (*i.e.*, 20 s for 30 *fps* video rate). In addition, in these *short-term* benchmarks, the target objects almost always appear in the video view. As a consequence, the evaluations on such *short-term* benchmarks may not reflect the performance of an algorithm in the real world, and thus restrict applications.

Limited number of object categories To assess the performance of tracking algorithms in the real world, it is necessary to utilize a diverse set of object categories for evaluation. However, most existing benchmarks contain less than 30 object categories for evaluation (see Fig. 2). In addition, these benchmarks do not provide any unseen object classes in evaluation, which makes it difficult to fully evaluate the tracking performance in real applications. We note that the recent GOT-10k (Huang et al. 2019) tackles this problem by introducing a large set of object classes for tracking.

Category bias A robust tracker should demonstrate stable performance in locating arbitrary targets regardless of their categories, which requires that *category bias* (or *class imbalance*) should be eliminated in training and/or evaluating tracking algorithms. Despite this, most current tracking benchmarks usually consist of a few object classes (see Table 1). The GOT-10k (Huang et al. 2019) alleviates the problem of category bias to some extent by introducing diverse categories. However, categories are not rigorously balanced as the number of videos varies a lot across different categories.

Evaluation for unseen category For certain applications (*e.g.*, tracking rare object classes with very few videos for training), it is desired to evaluate the performance of a

Table 1 Comparison of LaSOT with the most popular dense benchmarks in the literature. "Eva." and "Tra." indicate evaluation and training, respectively

Benchmark	OTB-13 (Wu et al. 2013)	OTB-15 (Wu et al. 2015)	TC-128 (Liang et al. 2015)	CDTB (Lukezic et al. 2019)	VOT-2017 (Kristan et al. 2017)	NUS-PRO (Li et al. 2016)	UAV123 (Mueller et al. 2016)	UAV20L (Mueller et al. 2016)	NfS (Galoogahi et al. 2017)	GOT-10k (Huang et al. 2019)	LaSOT
Num. of videos	51	100	128	80	60	365	123	20	100	9695	1550
Min frames	71	71	71	406	41	146	109	1717	169	29	1000
Mean frames	578	590	429	1274	356	371	915	2934	3830	149	2502
Median frames	392	393	365	1179	293	300	882	2626	2448	101	2145
Max frames	3872	3872	3872	2501	1500	5040	3085	5527	20,665	1418	11,397
Total frames	29**K**	59**K**	55**K**	102**K**	21**K**	135**K**	113**K**	59**K**	383**K**	1.45**M**	3.87**M**
Total duration	16.4 **m**	32.8 **m**	30.7 **m**	56.7 **m**	11.9 **m**	75.2 **m**	62.5 **m**	32.6 **m**	26.6 **m**	40 **h**	35.8 **h**
Video framerate	30 fps	30 fps	30 fps	30 fps	30 fps	30 fps	30 fps	30 fps	240 fps	10 fps	30 fps
Object classes	10	16	27	23	24	8	9	5	17	563*	85
Num. of attributes	11	11	11	13	n/a	n/a	12	12	9	6	14
Absent labels	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✓
Fully class balanced	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Axis alignment	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Lingual specification	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Full overlap protocol	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	✗	✓
One-shot protocol	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	✓	✓
Benchmark aim	Eva.	Eva.	Eva.	Eva.	Eva.	Eva.	Eva.	Eva.	Eva.	Tra./Eva.	Tra./Eva.

*Note that, GOT-10k (Huang et al. 2019) covers more specific object and motion classes. For example, 'person in jogging' and 'person in skiing' are treated as two different object classes. However, in this work (and all other benchmarks), both belong to the category of 'person'

tracker in locating targets belonging to previously unseen category. Current large-scale benchmark (*e.g.*(Müller et al. 2018)) often have category overlaps between training and testing sequences, which makes it hard to meet this evaluation requirement. In order to alleviate this problem, the recently proposed GOT-10k (Huang et al. 2019) that takes the first attempt to introduce one-shot evaluation for tracking, aiming to assess tracking performance for unseen object classes.

In the literature, many benchmarks have been introduced to handle the aforementioned problems: *e.g.*, (Wu et al. 2013; Liang et al. 2015; Kristan et al. 2016; Wu et al. 2015; Kristan et al. 2018; Li et al. 2016) for precise dense annotations, (Mueller et al. 2016; Valmadre et al. 2018) for long-term tracking, (Huang et al. 2019) for diverse object categories and unseen classes, and (Müller et al. 2018; Huang et al. 2019) for large-scale tracking. However, none of them address all of the issues, which motivates our proposed benchmark.

1.1 Contribution

In this work, we provide a novel benchmark for **La**rge-scale **S**ingle **O**bject **T**racking (LaSOT). The contributions of LaSOT are summarized as follows:

(1) We present a large-scale benchmark, LaSOT, for visual tracking. LaSOT covers 85 object categories and consists of 1550 videos totaling more than 3.87 M frames. Each frame is carefully inspected and manually labeled with a bounding box. To ensure quality, each annotation box is visually double-checked and corrected when needed. To our knowledge, LaSOT is by far the largest (in terms of the number of frames) tracking benchmark with precise dense annotations. By releasing LaSOT, we expect to offer the community a dedicated platform for unified training and evaluation of tracking algorithms.
(2) LaSOT allows evaluation of long-term tracking. In particular, the shortest sequence consists of 1000 frames and the longest 11,397 frames, and the average video length of LaSOT is around 2500 frames (equating to around 83 s, see Table 1), enabling assessment of long-term trackers.
(3) Different from current benchmarks which only provide bounding boxes, LaSOT offers both visual bounding box annotations and natural language specification, which has been shown to be beneficial for various vision tasks [*e.g.*, (Hu et al. 2016; Li et al. 2017)] including tracking (Li et al. 2017; Feng et al. 2020). By providing additional language annotations, we aim at stimulating the use of lingual features to further improve tracking.
(4) For flexible evaluation of trackers in different settings, we adopt two protocols, *i.e.*, *full overlap* and *one-shot*. For full overlap protocol, training and testing sets have the same object classes. For one-shot protocol, as introduced in (Huang et al. 2019), the categories of training and testing sets instead have zero overlap. These two protocols enable researchers/engineers to more flexibly evaluate their trackers to differing requirements, *e.g.*, locating targets belonging to seen/unseen categories.
(5) LaSOT inhibits category bias by collecting equal numbers of videos for each object class[1]. By doing this, the evaluation and comparison of trackers becomes more fair. To our knowledge, LaSOT is the first benchmark rigorously balanced for equal category size.
(6) To evaluate existing trackers and enable future comparison on LaSOT, we benchmark 48 representative tracking algorithms under the two protocols, and conduct extensive and in-depth analysis on performance using different metrics.

This paper extends an early conference version in (Fan et al. 2019). The main new contributions are follows. (1) We introduce 15 extra new object classes with 150 manually annotated sequences and more than 350 K frames. In particular, different from classes chosen based on ImageNet in (Fan et al. 2019), the 15 new classes are intentionally and carefully selected outside of ImageNet. By doing so, our benchmark enables a new *one-shot* evaluation protocol using these 15 classes for testing. (2) More details of benchmark construction are provided. (3) We employ two different protocols, *full overlap* and *one-shot*, for flexible performance evaluation for seen/unseen target categories. (4) More thorough experimental analysis are conducted in various aspects.

The rest of this paper is organized as follows. Section 2 discusses related tracking algorithms and datasets of this work. In Sect. 3, we detail the construction of LaSOT and analyze it through a variety of informative statistics. Experimental evaluation with in-depth analysis are conducted in Sect. 4, followed by conclusion in Sect. 5.

2 Related Work

2.1 Visual Tracking Algorithm

Visual tracking has been extensively studied in the past few decades. Here we briefly review two recent trends including correlation-filter trackers and deep trackers, and refer readers to surveys (Li et al. 2013; Smeulders et al. 2014; Yilmaz et al. 2006; Li et al. 2018) for more algorithms.

[1] Note that for tracking benchmark using *full overlap* split protocol, category bias should be inhibited in both training and evaluation of trackers. For tracking benchmark using *one-shot* split protocol, category bias should be inhibited in only training of trackers.

Correlation-filter approaches formulate tracking task as a regression problem by learning a discriminative filter. Owing to the extremely efficient solution using fast Fourier transform (FFT), correlation-filter trackers (Bolme et al. 2010; Henriques et al. 2015) run at speeds of several hundred frames per second and draw extensive attention with many improvements. The methods of Li and Zhu (2014); Danelljan et al. (2014) introduce a scale embedding to handle scale variation. The approaches in Danelljan et al. (2015); Li et al. (2018) improve correlation-filter tracking using extra regularization techniques. Background information is explored in Mueller et al. (2017); Galoogahi et al. (2017) to enhance robustness of the filters. The methods of Ma et al. (2015); Danelljan et al. (2016, 2017) replace hand-crafted features with deep features to improve performance. The approach in Liu et al. (2015) utilizes part-based representation to deal with challenges that are difficult for correlation filter tracking.

Inspired by the success of deep learning, many deep trackers (Wang and Yeung 2013; Wang et al. 2015; Nam and Han 2016; Fan and Ling 201; Song et al. 2018) have been proposed and exhibit state-of-the-art performance. Despite impressive results, these approaches suffer from heavy computational burden due to deep feature extraction or online network fine-tuning. To alleviate this problem, deep Siamese networks have been introduced for object tracking (Bertinetto et al. 2016b; Tao et al. 2016). Owing to balanced efficiency and accuracy, deep Siamese tracking has been extended by many later works (He et al. 2018; Li et al. 2018; Fan and Ling 2019; Wang et al. 2019, ?; Zhu et al. 2018; Li et al. 2019). To deal with scale variation, the methods of Danelljan et al. (2019) introduce the intersection-over-union (IoU) network for tracking and achieve promising results.

2.2 Visual Tracking Benchmark

Benchmarks have been crucial for advancing the research in visual tracking. For a systematic review, we classify existing benchmarks into two types: *dense benchmarks* which use per-frame manual annotation and *other benchmarks* which use sparse and/or (semi-)automatic annotation.

2.2.1 Dense Benchmarks

Dense benchmarks offer *per-frame* bounding box annotations for each video. In order to ensure high quality, each frame is manually annotated with careful inspection and verification. For tracking, these precise bounding box annotations are highly desired for both training and evaluating tracking algorithms. Currently, popular dense tracking benchmarks include OTB (Wu et al. 2013, 2015), TC-128 (Liang et al. 2015), VOT (Kristan et al. 2016), NUS-PRO (Li et al. 2016), UAV (Mueller et al. 2016), NfS (Galoogahi et al. 2017), CDTB (Lukezic et al. 2019) and GOT-10k (Huang et al. 2019).

OTB OTB-13 (Wu et al. 2013) contains 51 videos with manual annotation for tracking evaluation. The videos are labeled with 11 attributes for further analysis of tracking performance. OTB-13 was later extended to the larger OTB-15 (Wu et al. 2015) by introducing extra 50 sequences.

TC-128 TC-128 (Liang et al. 2015) comprises of 128 videos that are specifically designated to evaluate color-enhanced trackers. The videos are labeled with 11 similar attributes as in OTB (Wu et al. 2013).

VOT VOT (Kristan et al. 2016) introduces a series of tracking competitions with up to 60 sequences in each of them, aiming to evaluate the performance of a tracker in a relative short duration. Each frame in the VOT datasets is annotated with a rotated bounding box with several attributes.

CDTB CDTB (Lukezic et al. 2019) offers 80 RGB-D videos with manual annotations for tracking. Each sequence is labeled with 13 attributes. The goal of CDTB is to encourage the exploration of depth information for improving tracking performance.

NUS-PRO NUS-PRO (Li et al. 2016) contains 365 sequences with a focus on human and rigid object tracking. Each sequence in NUS-PRO is annotated with both target location and occlusion level for evaluation.

UAV UAV123 and UAV20L (Mueller et al. 2016) are utilized for unmanned aerial vehicle (UAV) tracking, comprising 123 short and 20 long sequences, respectively. Both UAV123 and UAV20L are labeled with 12 attributes.

NfS NfS (Galoogahi et al. 2017) provides 100 sequences with a high frame rate of 240 fps, aiming to analyze the effects of appearance variations on tracking performance.

GOT-10k GOT-10k (Huang et al. 2019) consists of 9695 videos, aiming to provide rich motion trajectories for developing and evaluating trackers. In addition, GOT-10k is the first to propose a novel one-shot evaluation for assessing tracking performance.

Our LaSOT belongs to the category of dense tracking benchmark. In comparison with others, LaSOT is the *largest* with more than 3.87 million frames and an average video length of around 2500 frames. Moreover, LaSOT is the only one to offer additional language specification for each sequence. LaSOT is closely related to but different from the recently proposed large-scale GOT-10k (Huang et al. 2019). Despite sharing the similar idea of performing one-shot evaluation, LaSOT presents two protocols. In addition, instead of focusing on short-term tracking GOT-10k, our goal is to assess trackers in long-term scenarios. Table 1 provides a

detailed comparison of LaSOT with existing dense benchmarks. It is worth noting that most existing dense tracking benchmarks, including LaSOT, utilize axis-aligned bounding boxes to annotate targets. The reasons are two-fold. First, the problem setting of current single object tracking is to locate the target with a manually given up-right bounding box. In accordance with this goal, axis-aligned boxes are usually adopted to annotate targets in many benchmarks. Axis-aligned boxes are also widely employed in object detection benchmarks such as PASCAL VOC (Everingham et al. 2010) and COCO (Lin et al. 2014). Second, axis-aligned boxes are able to provide sufficient information about the target for stable tracker initialization and reliable performance evaluation, as evidenced by recent progresses of tracking algorithms on various benchmarks. From this perspective, axis-aligned boxes are effective for tracking. Moreover, this type of annotation requires less labeling efforts.

2.2.2 Other Tracking Benchmarks

Aside from benchmarks described above, there are other benchmarks using different annotation strategies. These tracking benchmarks are either labeled sparsely (*e.g.*, every 30 frames) or annotated (semi)-automatically using tracking algorithms. Examples of these types of benchmarks include ALOV (Smeulders et al. 2014), TrackingNet (Müller et al. 2018) and OxUvA (Valmadre et al. 2018).

ALOV (Smeulders et al. 2014) comprises of 314 video sequences which are labeled in 14 attributes. Instead of per-frame annotation, ALOV provides annotations every 5 frames. **TrackingNet** (Müller et al. 2018) is a large-scale benchmark with 30K sequences. All videos come from the video object detection dataset YT-BB (Real et al. 2017), and each one is labeled by a tracking algorithm. Although this tracker annotator is shown to be reliable in a relatively short period (*i.e.*, 1 s), it is hard to guarantee the same tracking performance on a different benchmark, especially when the sequences become more challenging. In addition, the average video length of TrackingNet is less than 500 frames, which may not be able to reflect the long-term performance of a tracking algorithm. **OxUvA** (Valmadre et al. 2018) consists of 366 sequences. Similar to TrackingNet, the videos are sampled from YT-BB (Real et al. 2017). With the average sequence length more than 4200 frames, OxUvA mainly aims to focus on long-term tracking. Each video in OxUvA is labeled every 30 frames.

These benchmarks usually provide a large number of sequences and serve well for evaluation purposes. While they benefit from a reduction of annotation cost, they do not provide detailed per frame performance evaluation of tracking algorithms. Furthermore, it may cause problems for some trackers that require temporal context or motion cues from annotations, because these information may be either missing due to sparse annotation or imprecise due to potentially unreliable annotation. Different from these benchmarks, LaSOT provides a large set of sequences with high-quality dense bounding box annotations, which makes it more suitable for developing deep trackers as well as for evaluating long-term tracking algorithms.

2.3 Other Vision Benchmarks

Given the similarities shared between visual object tracking and video object detection (*e.g.*, visual tracking can be treated as video single-object detection), video object detection benchmarks VID (Russakovsky et al. 2015) and YT-BB (Real et al. 2017) are often adopted for training deep trackers.

VID (Russakovsky et al. 2015) consists of 5.4 K sequences with more than two million frames and **YT-BB** (Real et al. 2017) contains 380 K videos with more than five million frames. Despite being large in scale, these two benchmarks are not ideally suitable for tracking tasks due to several reasons. First, in many videos, the targets are almost static throughout the entire video, making them not desirable for motion tracking. Second, the targets are partially out of view in the initial frame in a lot of videos, which is different from the tracking task. Third, the benchmarks are sparsely annotated, and thus may be inappropriate if directly used for tracking as discussed early.

In the era of deep learning, benchmarks have played a more important role in advancing various vision tasks. To some extent, LaSOT is inspired by the successes of other vision benchmarks. To this end, we will briefly discuss several large-scale benchmarks in other tasks including image classification, object detection, segmentation and multi-object tracking.

In image classification, **ImageNet** (Deng et al. 2009) is arguably the most popular dataset consisting of more than 10M images. Owing to the large-scale ImageNet, deep networks have proven their power in learning visual representation. In object detection, the well-known **PASCAL VOC** detection (Everingham et al. 2010) contains around 10 K images. The larger scale **COCO** (Lin et al. 2014) contains more than 200 K images for detection. In image segmentation, **PASCAL VOC** segmentation (Everingham et al. 2010) provides around 10 K images. **ADE20K** (Zhou et al. 2017) is a collection of more than 20 K images for scene parsing. **Citiscapes** (Cordts et al. 2016) consists of 25 K images for traffic scene segmentation. **LVIS** (Gupta et al. 2019) offers 164 K image for large-scale vocabulary instance segmentation. In multi-object tracking, the **MOT** challenge (Milan et al. 2016) provides 21 videos. Recently, a larger scale **TAO** (Dave et al. 2020) has been compiled containing 2907 videos.

3 The LaSOT Benchmark

3.1 Design Principle

Our goal is to construct a dedicated benchmark, LaSOT, for training and evaluating tracking algorithms. To this end, we follow six principles in constructing LaSOT, including *large-scale*, *high-quality dense annotations*, *long-term tracking*, *category balance*, *comprehensive labeling* and *flexible protocols*, aimed at handling the issues of existing tracking benchmarks described in previous sections.

3.2 Data Collection

In total, LaSOT consists of 85 object classes, which are divided into two parts. The first part, referred to as *part-1* for short, contains 1400 sequences from 70 object categories. Most of categories are chosen from the 1000 classes from ImageNet (Deng et al. 2009), with a few exceptions (*e.g.*, *drone*) that are carefully selected for popular tracking applications. The other part, referred to as *part-2* for short, comprises 150 sequences from 15 object classes. It is worth noting that, for the goal of one-shot evaluation on object from unseen categories, these 15 classes are carefully chosen from ***outside*** object categories in ImageNet (Deng et al. 2009) and intentionally to be far away from the 70 categories in part-1. There is no overlap between the 15 categories in part-2 and 70 classes in part-1. Different from current dense tracking benchmarks that contain less than 30 categories and typically are unevenly distributed, LaSOT provides equal number of videos for each category in both part-1 and part-2 to avoid the category bias problem.

After determining the 85 object classes in LaSOT, we searched for sequences of each category from YouTube (https://www.youtube.com/). The reasons for choosing YouTube are two-fold: (1) YouTube is the largest video platform in the world, which allows us to select diverse videos for constructing the benchmark and avoids bias to certain scenes, and (2) many videos on YouTube are captured in the wild, which may be helpful for developing and evaluating trackers for real applications.

Initially, over 6000 video sequences are collected. With a joint consideration of the video quality (*e.g.*, videos with shot cut are not suitable for tracking) and our design principles, 1550 sequences survived. Nevertheless, these 1550 videos are not immediately available for the tracking task due to containing a large mount of irrelevant contents. For instance, for a video of *person* category (*e.g.*, a sporter), it often consists of some undesirable introduction content of each sporter in the beginning. Therefore, we carefully inspect each video sequence, filter out the tracking-unrelated contents and exclusively retain one usable clip for our tracking task. For part-1, each category consists of 20 videos, while for part-2, each contains 10 sequences. Figure 3 shows the object categories on LaSOT with comparison to several existing popular dense tracking benchmarks with available category information. It is worth noting that, although the numbers of videos for categories in part-1 and part-2 are not equal, LaSOT is still balanced due to their different roles as described in Sect. 3.5. Also note that, in Fig. 3 we do not include the large-scale GOT-10k for comparison because the category granularity used in GOT-10k is different from those in other benchmarks. For example, "big truck ", "half truck" and "pickup truck" are treated as three different categories in GOT-10k. By contrast, in other benchmarks, there may exist only one "truck" category.

Eventually, we compiled a large-scale tracking benchmark by gathering 1550 videos with 3.87 million frames from YouTube under Creative Commons license. The average video length of LaSOT is 2502 frames (*i.e.*, 83 s for 30 fps). The shortest sequence contains 1000 frames (*i.e.*, 33 s), while the longest one consists of 11,397 frames (*i.e.*, 378 s).

3.3 Annotation

3.3.1 Annotation Protocol

Annotation consistency cross different sequences and labelers is crucial for the quality of a tracking benchmark. We define a deterministic protocol for ensuring such quality. In a video sequence with a specific tracking target (determined before starting annotation), for each frame, if the target is present in the view, a labeler manually draws/edits an upright (axis-aligned) bounding box to tightly fit any visible part of the target (see left images of (a) and (b) in Fig. 4); otherwise, an absence label, either *full occlusion* (see right image of (a) in Fig. 4) or *out-of-view* (see right image of (b) in Fig. 4), is assigned to this frame. By doing so, there are two advantages: (1) with absence labels, performance evaluation is more accurate by avoiding those frames without target present, and (2) researchers can develop occlusion or out-of-view aware tracking algorithms using this information. Note that, our strategy cannot guarantee to minimize the background area in the box, as similarly observed in other benchmarks. Nevertheless, this strategy provides consistent annotations that are relatively stable for learning the dynamics.

The above annotation strategy works well most of the time, however, exceptions exist. For certain categories, *e.g.*, *mouse*, the target object may contain long, thin, and/or highly deformable parts, *e.g.*, a tail, which not only introduces much background information into object, but also provides little help for target recognition and localization. We carefully identify such targets and associated videos in LaSOT, and design specific rules for their annotations. In detail, before

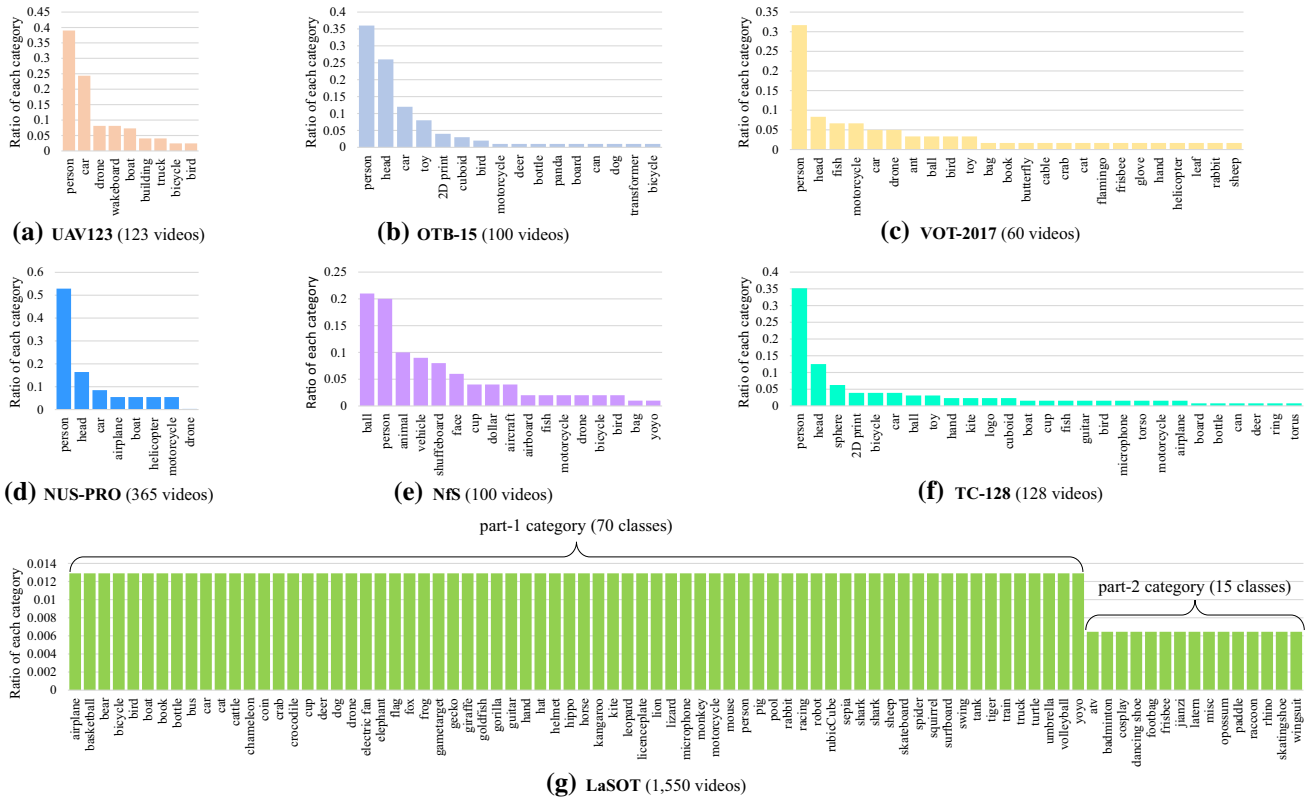


Fig. 3 Category distribution of tracking benchmarks. The category distribution of LaSOT is more balanced than those of other benchmarks. Best viewed in color

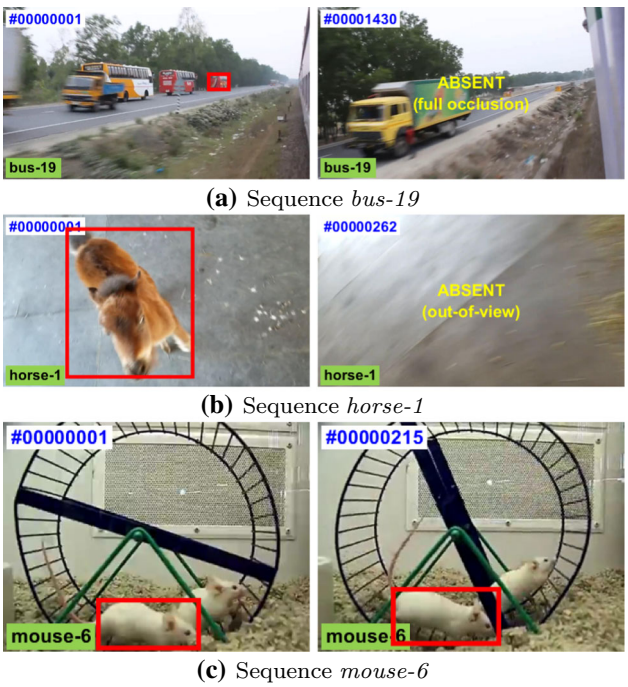


Fig. 4 Illustration of annotation strategy for different cases. **a**: target with full occlusion. **b**: target with out-of-view. **c**: target with thin, long, and highly deformable part. Best viewed in color

starting to annotate, we inspect each object category and identify twelve such categories and their undesired parts, including *bird* (the *leg* part), *cat* (the *tail* part), *elephant* (the *tail* part), *fox* (the *tail* part), *gecko* (the *tail* part), *guitar* (the *handlebar* part), *leopard* (the *tail* part), *lion* (the *tail* part), *monkey* (the *tail* part), *tiger* (the *tail* part), *shark* (the *tail* part) and *mouse* (the *tail* part). For objects from these categories, we exclude the undesired part when drawing their bounding boxes. Note that, to ensure the usability of these classes, the inspection of each object category and identification of undesired parts are conducted by a group of experts (three PhD students working in related areas). An annotation example is shown in the image (c) of Fig. 4 and more can be found in our benchmark.

In order to enrich annotation, we provide additional language descriptions for each sequence. The natural language specification is represented by a sentence that describes the color, behavior and surroundings of the target. LaSOT consists of 1550 such sentences for all sequences. Notice that, we expect that these lingual descriptions can provide auxiliary help for improving tracking. For example, one can leverage deep neural networks to extract lingual features and use them as a global semantic guidance to suppress background distractors in the search region. This way, the tracker may better focus on locating the target.

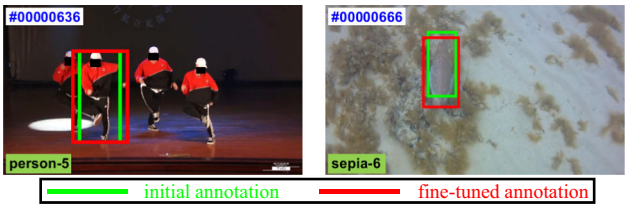


Fig. 5 Examples of fine-tuning initial annotations. We observe that, after fine-tuning, the final annotations in red rectangles better fit to the target region than the initial annotations in green rectangle. Best viewed in color

3.3.2 Quality Assessment Protocol

For developing a high-quality dense benchmark, the most effort demanding parts include *manual labeling*, *double-checking* and *error correcting*. For this task, we have assembled an annotation team composed of several Ph.D. students working on related areas and many volunteers. To ensure high-quality annotation, each video is processed by two teams: a labeling team and a validation team. Each labeling team is composed of a volunteer and an expert (PhD student). The volunteer manually draws/edits the target bounding box in each frame, and the expert inspects the results and adjusts them if necessary. Then, the annotation results are reviewed by the validation team composed of several (typically three) experts. If an annotation result is not unanimously agreed by all members in the validation team, it will be sent back to the original labeling team to revise. Note that, when sending the annotation results back for revision, detailed comments from the validation team are attached. Examples include "the annotated bounding box is too small to cover the whole target," "the box is too large and introduce too much background," "the left side contains much background and its edge needs to move closer the target boundary," etc. This way, we ensure that the revised annotation result is acceptable as expected.

To improve the annotation quality as much as possible, we check all the annotation results carefully and revise them frequently. Around 40% of the initial annotations fail in the first round of validation, and many frames are revised at least three times. Some challenging frames that are initially labeled incorrectly or inaccurately are given in Fig. 5. With all these efforts, we finally reach a benchmark with high-quality dense annotation, with some examples shown in Fig. 6.

3.4 Attributes

In order to further analyze the tracking performance, each sequence in LaSOT is labeled with a list of 14 attributes, including camera motion (CM), rotation (ROT), deformation (DEF), full occlusion (FOC), partial occlusion (POC), illumination variation (IV), out-of-view (OV), viewpoint change (VC), scale variation (SV), background clutter (BC), motion blur (MB), aspect ratio change (ARC), low resolution (LR) and fast motion (FM). Table 2 lists the definition of each attribute, and Fig. 7a shows the distribution of sequences in each attribute. From Fig. 7a, it can be seen that the most common challenge factors in LaSOT are target scale changes (SV and ARC), occlusion (POC and FOC), deformation and rotation, which frequently occur in real applications.

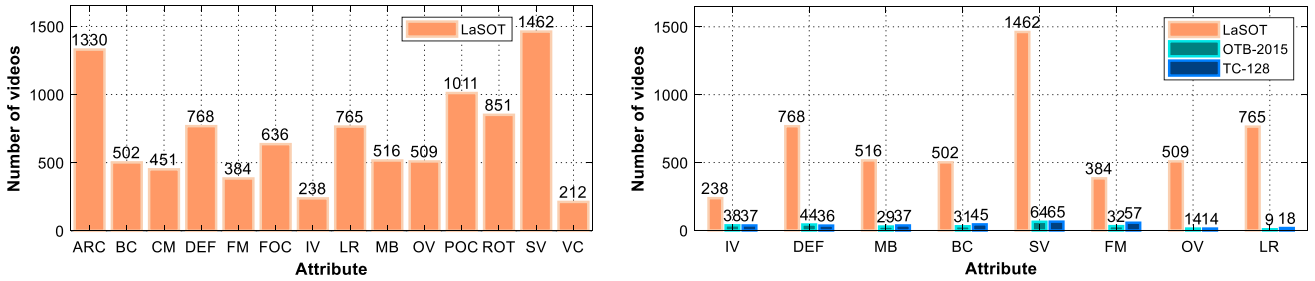
In addition, Fig. 7a shows that each attribute consists of *at least* 200 videos, which clearly supports the statistical significance of our attribute evaluation. Figure 7b demonstrates the distribution of attributes of LaSOT compared with popular benchmarks OTB-15 (Wu et al. 2015) and TC-128 (Liang et al. 2015) on overlapping attributes. From Fig. 7b, we observe that more than 1400 videos in LaSOT are involved with scale variations. Compared with OTB-2015 and TC-128 with less than 70 videos with scale changes, LaSOT is more challenging and thus better reflects the generalizability of trackers in dealing with scale changes. On the out-of-view attribute, LaSOT contains 509 videos, while OTB-15 and TC-128 have less than 20 sequences, indicating that LaSOT reflects better the challenges for tracking in the wild. Moreover, LaSOT focuses on small object tracking with 765 videos in the attribute of low resolution, much more than that in OTB-15 and TC-128.

It is worth noting that in our benchmark, as well as in most other popular ones, a video sequence may consist of more than one attribute. As a consequence, it may be difficult to concretely identify the attribute causing failure, especially if the number of videos on this attribute for evaluation is small. The ideal situation for attribute evaluation would be that each sequence exhibits *one and only one* attribute. Nevertheless, in real world applications, it is almost impossible for a video to contain only one challenge. To alleviate this problem and reduce uncertainty, in existing tracking benchmarks, a common way is to collect all videos containing a specific attribute when performing evaluation for that attribute. For example, one can usually gather more than thirty videos for some attributes. Especially, in our large-scale benchmark, the numbers of videos for most attributes exceed one hundred. Consequently, we may obtain a *statistically meaningful* conclusion for attribute evaluation despite that videos may contain mixed attributes. This is supported by the fact that many trackers with higher attribute evaluation scores generally work better in dealing with corresponding attributes in videos on various benchmarks. For this reason, following the studies in previous tracking benchmarks, attribute-based evaluation is conducted on LaSOT as well. With that said, it is worth noting a recent effort to restrict one attribute per (short) sequence in tracking evaluation (Fan et al. 2020).

Fig. 6 Example sequences and annotations in LaSOT. Best viewed in color

Table 2 Descriptions of 14 different attributes in LaSOT

Attribute	Definition
CM	*Camera Motion*: abrupt motion of the camera
VC	*View Change*: viewpoint affects target appearance significantly
ROT	*Rotation*: the target object rotates in the image
SV	*Scale Variation*: the ratio of target bounding box is outside the range [0.5, 2]
DEF	*Deformation*: the target object is deformable during tracking
BC	*Background Clutter*: the background near the target object has the similar appearance as the target
POC	*Partial Occlusion*: the target object is partially occluded in the sequence
FOC	*Full Occlusion*: the target object is fully occluded in the sequence
MB	*Motion Blur*: the target region is blurred due to the motion of target object or camera
IV	*Illumination Variation*: the illumination in the target region changes
ARC	*Aspect Ratio Change*: the ratio of bounding box aspect ratio is outside the rage [0.5, 2]
OV	*Out-of-View*: the target object completely leaves the video frame
LR	*Low Resolution*: the area of target box is smaller than 1000 pixels in at least one frame
FM	*Fast Motion*: the motion of target object is larger than the size of its bounding box



(a) Distribution of sequences in each attribute on LaSOT

(b) Comparison of distribution in common attributes on different benchmarks

Fig. 7 Distribution of sequences in each attribute in LaSOT and comparison with other benchmarks. Best viewed in color

Table 3 Comparisons between training/testing sets of LaSOT under **full overlap** protocol

	Video	Min frames	Mean frames	Max frames	Total frames
$\text{LaSOT}_{\text{tra}}$	1120	1000	2529	11,397	2.83 **M**
$\text{LaSOT}_{\text{tst}}$	280	1000	2448	9999	690 **K**

Table 4 Comparisons between training/testing sets of LaSOT under **one-shot** protocol

	Video	Min frames	Mean frames	Max frames	Total frames
$\text{LaSOT}_{\text{tra}}$	1400	1000	2506	11,397	3.52 **M**
$\text{LaSOT}_{\text{tst}}$	150	2005	2393	2500	350 **K**

3.5 Evaluation Protocols

Currently, evaluation of large-scale benchmarks is based on either *full overlap* (*e.g.*, Müller et al. (2018)) or *one-shot* (*e.g.*, Huang et al. (2019)). We argue that both protocols have their own applications. The full overlap protocol splits training/testing sets with fully overlapped object classes, and it can be used to develop tracking algorithms in the scene where the target category appears in the tracker's training set. By contrast, one-shot protocol splits training/testing with no overlap between their object categories, and it can be utilized in applications where the target category is rare. In order to accommodate more application scenarios, we introduce both protocols into LaSOT.

Full Overlap Protocol In the full overlap protocol, 1400 sequences of 70 categories in part-1 are used for training and testing. Specifically, following the 80/20 principle (*i.e.*, the Pareto principle), we select 16 out of 20 sequences in each category for training, and the rest for testing. This way in the full overlap protocol, the training and testing sets consist of 1120 and 280 videos respectively. Since the number of videos in each category for both training and testing are equal, LaSOT is category-balanced. Table 3 compares statistics of training/testing sets in full overlap protocol.

One-shot Protocol In the one-shot protocol, all 1550 videos from the 85 classes are utilized for training and testing. Because training and testing sets are required to have no overlap in category, we employ 1400 sequences of 70 categories in part-1 for training, and the other 150 videos of 15 classes in part-2 are used for evaluation. In particular, to increase the source difference, the 15 objects categories are specially chosen outside of the 1000 classes from ImageNet. It is worth noting that LaSOT is still category-balanced because in both sets, each category contains the same number of videos. Table 4 compares statistics of training/testing sets in one-shot protocol.

4 Evaluation

4.1 Evaluation Metric

Following Wu et al. (2015), we perform One-Pass Evaluation (OPE) and measure the performance of different trackers using three metrics, *i.e.*, **precision**, **normalized precision** and **success**, under two protocols.

The precision (**PRE**) is calculated by comparing distance between centers of the groundtruth bounding box and the tracking result in pixels. Different algorithms are ranked according to the value of this metric on a certain threshold (*e.g.*, typically 20 pixels). Since PRE does not take object scale into consideration, it is sensitive to target size and image resolution. To avoid this problem, we adopt an additional strategy as in Müller et al. (2018) to normalize the PRE with scales. Please refer to Müller et al. (2018) for more details.

Table 5 Summary of evaluated trackers. Representation: Sparse — Sparse Representation, Color — Color Names or Histograms, Pixel — Pixel Intensity, HoG — Histogram of Oriented Gradients, H or B — Haar or Binary, Deep — Deep Features, Update — Online model update (or fine-tuning)

		Representation								Search		
		PCA	Sparse	Color	Pixel	HoG	H or B	Deep	Update	PF	RS	DS
IVT (Ross et al. 2008)	IJCV08	✓							✓	✓		
MIL (Babenko et al. 2009)	CVPR09						H		✓			✓
Struck (Hare et al. 2011)	ICCV11						H		✓			✓
L1APG (Bao et al. 2012)	CVPR12		✓						✓	✓		
ASLA (Jia et al. 2012)	CVPR12		✓						✓	✓		
CSK (Henriques et al. 2012)	ECCV12				✓				✓			✓
CT (Zhang et al. 2012)	ECCV12						H		✓			✓
TLD (Kalal et al. 2012)	PAMI12						B		✓			✓
CN (Danelljan et al. 2014)	CVPR14			✓	✓				✓			✓
DSST (Danelljan et al. 2014)	BMVC14				✓	✓			✓			✓
MEEM (Zhang et al. 2014)	ECCV14				✓				✓		✓	
STC (Zhang et al. 2014)	ECCV14				✓				✓			✓
SAMF (Li and Zhu 2014)	ECCVW14			✓	✓	✓			✓			✓
LCT (Ma et al. 2015)	CVPR15				✓	✓			✓			✓
SRDCF (Danelljan et al. 2015)	ICCV15					✓			✓			✓
HCFT (Ma et al. 2015)	ICCV15							VGG-19	✓			✓
KCF (Henriques et al. 2015)	PAMI15					✓			✓			✓
Staple (Bertinetto et al. 2016a)	CVPR16			✓		✓			✓			✓
SINT (Tao et al. 2016)	CVPR16							VGG-16			✓	
SCT4 (Choi et al. 2016)	CVPR16					✓			✓			✓
MDNet (Nam and Han 2016)	CVPR16							VGG-M	✓		✓	
SiamFC (Bertinetto et al. 2016b)	ECCVW16							AlexNet				✓
Staple_CA(Mueller et al. 2017)	CVPR17			✓		✓			✓			✓
ECO_HC (Danelljan et al. 2017)	CVPR17					✓			✓			✓
ECO (Danelljan et al. 2017)	CVPR17							VGG-M	✓			✓
CFNet (Valmadre et al. 2017)	CVPR17							AlexNet	✓			✓
CSRDCF (Lukezic et al. 2017)	CVPR17			✓	✓	✓			✓			✓
PTAV (Fan and Ling 2017)	ICCV17				✓	✓		VGG-16	✓			✓
DSiam (Guo et al. 2017)	ICCV17							AlexNet				✓
BACF (Galoogahi et al. 2017)	ICCV17					✓			✓			✓
fDSST (Danelljan et al. 2017)	PAMI17				✓	✓			✓			✓
VITAL (Song et al. 2018)	CVPR18							VGG-M	✓		✓	
TRACA (Choi et al. 2018)	CVPR18							VGG-M	✓			✓
STRCF (Li et al. 2018)	CVPR18					✓			✓			✓
D-STRCF (Li et al. 2018)	CVPR18							VGG-M	✓			✓
StructSiam (Zhang et al. 2018)	ECCV18							AlexNet				✓
DaSiamRPN (Zhu et al. 2018)	ECCV18							Res-50	✓			✓
SiamRPN++ (Li et al. 2019)	CVPR19							Res-50				✓
SiamDW (Zhang and Peng 2019)	CVPR19							Res-22				✓
SiamMask (Wang et al. 2019)	CVPR19							Res-50				✓
ASRCF (Dai et al. 2019)	CVPR19					✓		VGG-16	✓			✓
ATOM (Danelljan et al. 2019)	CVPR19							Res-18	✓			✓
C-RPN (Fan and Ling 2019)	CVPR19							AlexNet				✓
GFSDCF (Xu et al. 2019)	ICCV19							Res-50	✓			✓

Table 5 continued

		Representation								Search		
		PCA	Sparse	Color	Pixel	HoG	H or B	Deep	Update	PF	RS	DS
DiMP (Bhat et al. 2019)	ICCV19							Res-50	✓			✓
SPLT (Yan et al. 2019)	ICCV19							Res-50	✓			✓
GlobalTrack (Huang et al. 2020)	AAAI20							Res-50				✓
LTMU (Dai et al. 2020)	CVPR20							Res-50	✓			✓

Search: *PF* Particle filter, *RS* Random sampling, *DS* Dense sampling

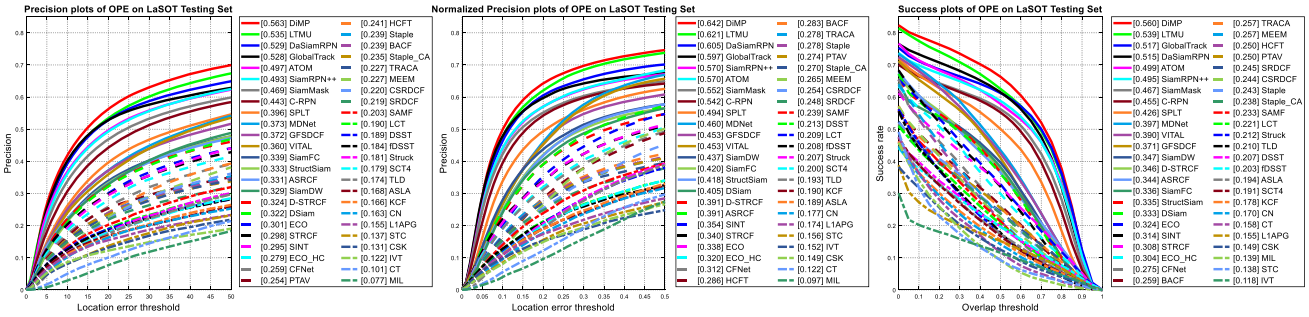


Fig. 8 Overall evaluation results on LaSOT under the *full overlap* protocol. Best viewed in color

The resulted normalized precision (**N-PRE**) can ensure the consistency of evaluation across different target scales. The success rate (**SUC**) is computed as the ratio of the number of successfully tracked frames (*i.e.*, intersection-over-union (IoU) between groundtruth bounding box and tracking result larger than a pre-defined threshold, typically, 0.5) to the number of all frames in a sequence.

4.2 Evaluated Tracking Algorithms

In order to provide baselines for future comparison on LaSOT, we extensively evaluate 48 algorithms. In specific, these 48 approaches consist of deep trackers (*e.g.*, MDNet Nam and Han (2016), TRACA Choi et al. (2018), CFNet Valmadre et al. (2017), SiamFC Bertinetto et al. (2016b), StructSiam Zhang et al. (2018), DSiam Guo et al. (2017), SINT Tao et al. (2016), ATOM Danelljan et al. (2019), DiMP Bhat et al. (2019), VITAL Song et al. (2018), SiamRPN++ Li et al. (2019), DaSiamRPN Zhu et al. (2018), SiamDW Zhang and Peng (2019), C-RPN Fan and Ling (2019) and SiamMask Wang et al. (2019), GlobalTrack Huang et al. (2020)), correlation trackers with hand-crafted features (*e.g.*, ECO_HC Danelljan et al. (2017), DSST Danelljan et al. (2014), CN Danelljan et al. (2014), CSK Henriques et al. (2012), KCF Henriques et al. (2015), fDSST Danelljan et al. (2017), SAMF Li and Zhu (2014), SCT4 Choi et al. (2016), STC Zhang et al. (2014) and Staple Bertinetto et al. (2016a)) or deep features (*e.g.*, HCFT Ma et al. (2015), D-STRCF Li et al. (2018), ECO Danelljan et al. (2017), GFSDCF Xu et al. (2019), ASRCF Dai et al. (2019)) and regularization techniques (*e.g.*, SRDCF Danelljan et al. (2015), STRCF Li et al. (2018), BACF Galoogahi et al. (2017), Staple_CA Mueller et al. (2017) and CSRDCF Lukezic et al. (2017)), ensemble trackers (*e.g.*, SPLT Yan et al. (2019), LTMU Dai et al. (2020), PTAV Fan and Ling (2017), LCT Ma et al. (2015), MEEM Zhang et al. (2014) and TLD Kalal et al. (2012)), sparse trackers (*e.g.*, L1APG Bao et al. (2012) and ASLA Jia et al. (2012)), other representatives (*e.g.*, CT Zhang et al. (2012), IVT Ross et al. (2008), MIL Babenko et al. (2009) and Struck Hare et al. (2011)). In evaluation, each tracker is used as it is, without any modification. Table 5 summarizes these trackers with their representation schemes and search strategies in a chronological order.

Note that in our evaluation, each tracker is tested as it is in the original paper, for three reasons. First, each tracker may require different training strategy. As a consequence, it is difficult to optimally train all trackers to obtain the best performance. Moreover, inappropriate training settings may result in performance drop for certain trackers. Second, despite using different training data, most deep trackers, especially recently proposed ones, have been fully trained on multiple large scale benchmarks. It is reasonable to assume that each tracker has attained optimal or decent performance in the originally published paper. Third, for trackers that only employ pre-trained classification backbone for feature extraction, it is hard to fine-tune the feature backbone network using existing tracking benchmarks.

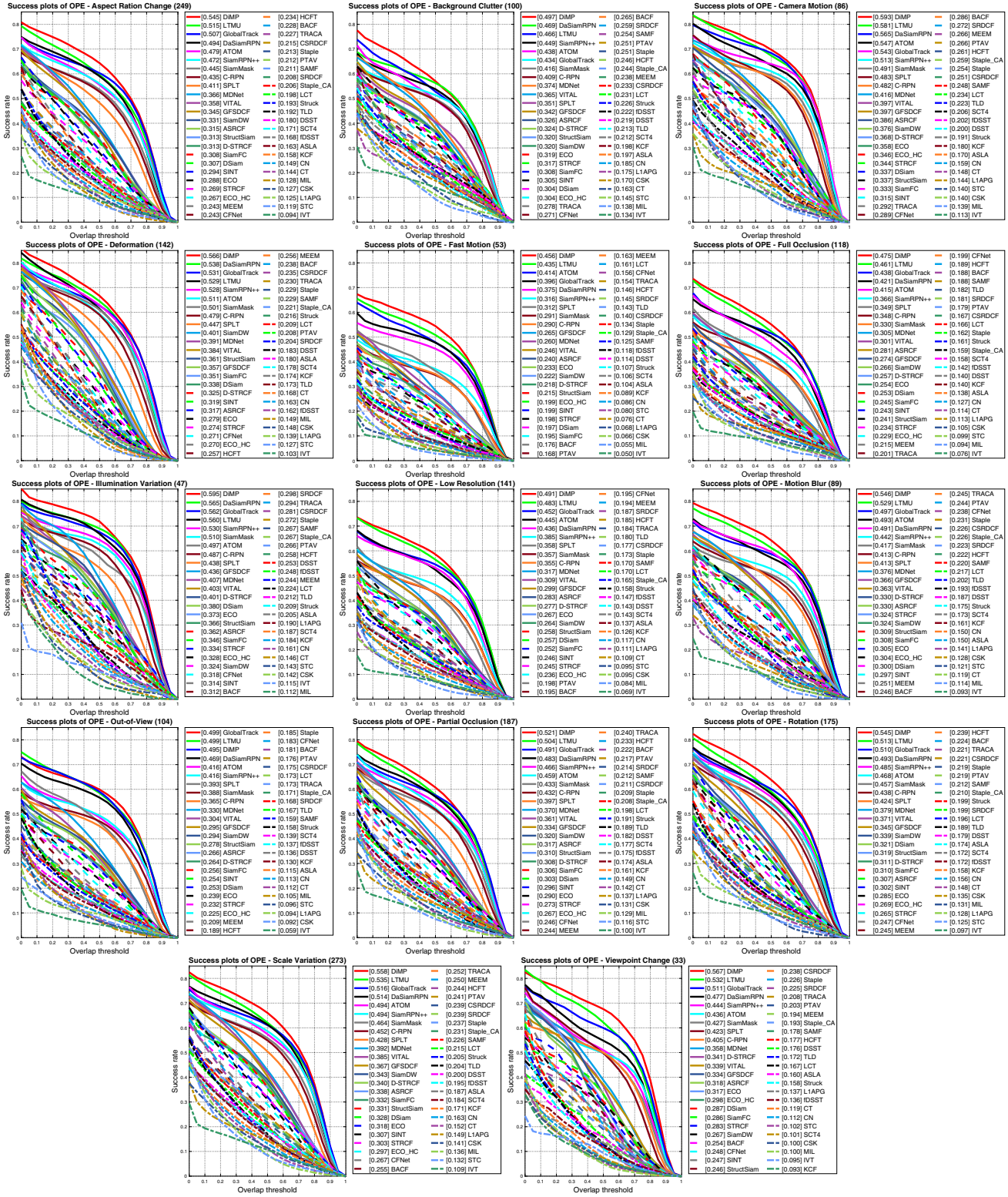


Fig. 9 Performance of trackers on each attribute using success under *full overlap* protocol. Best viewed in color

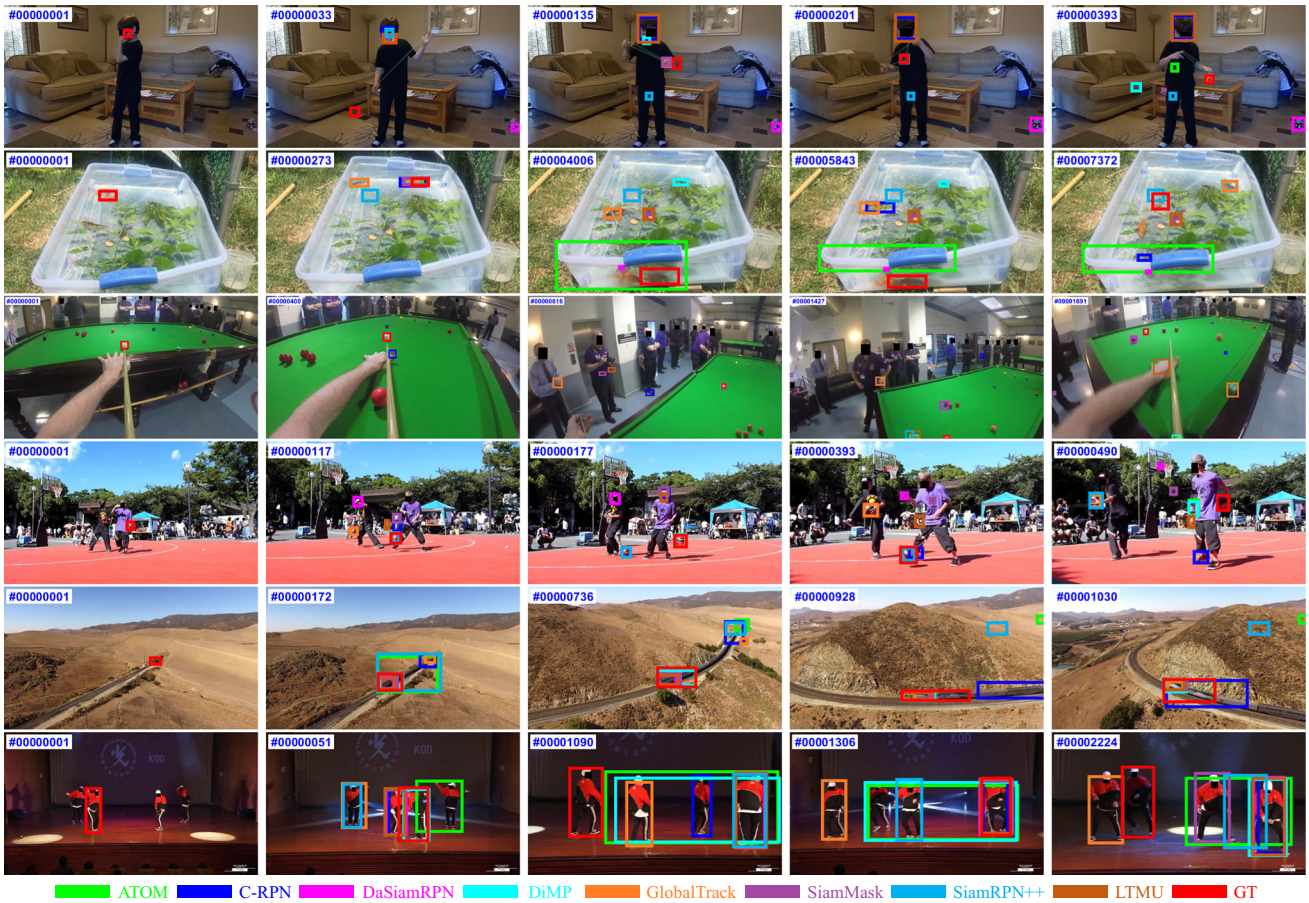


Fig. 10 Qualitative evaluation in six typically hard challenges in testing sequences of full overlap protocol (from top to bottom): *yoyo-7* with fast motion, *goldfish-7* with full occlusion, *pool-3* with low resolution, *basketball-11* with out-of-view, *train-1* with aspect ration change and *person-5* with background clutter. Best viewed in color

4.3 Evaluation with Full Overlap Protocol

4.3.1 Overall Performance

Figure 8 reports the evaluation results under full overlap protocol in OPE using precision (PRE), normalized precision (N-PRE) and success rate (SUC). DiMP achieves the best performance with PRE score of 0.563, N-PRE score of 0.642 and SUC score of 0.560. DiMP consists of two components including for target localization and scale estimation, both trained on a large set of videos. In addition, the target localization part is online updated during tracking. LTMU shows the second best performance with a 0.535 PRE score, 0.621 N-PRE score and 0.539 SUC score. LTMU focuses on long-term tracking by combining different components such as local tracker and detector. DaSiamRPN obtains the third best results with a 0.529 PRE score, 0.605 N-PRE score and 0.515 SUC score. This method is developed based on SiamRPN++ but utilizes more training data with augmentation techniques. Besides, a re-detection strategy and online model update are adopted for robust long-term tracking. Therefore, DaSiamRPN performs better than its baseline SiamRPN++ with a 0.493 PRE score, 0.57 N-PRE score and 0.495 SUC score. GlobalTrack introduces a two-stage framework for long-term tracking and demonstrates competitive results with a 0.528 PRE score, 0.597 N-PRE score and 0.517 SUC score. ATOM obtains promising results with a 0.497 PRE score, 0.57 N-PRE score and 0.499 SUC score. ATOM introduces an specific network to deal with scale variation. In addition, it employs complex method for optimization and acceleration to achieve real time speed. SiamFC tracker, which learns offline a matching function for tracking, achieves competitive results with a 0.339 PRE score, 0.42 N-PRE score and 0.336 SUC score. It is worth noticing that, unlike the performance on small benchmarks (*e.g.*, OTB-15 Wu et al. (2015)), SiamFC performs better than many more complicated algorithms such as StructSiam, DSiam, PTAV, and HCFT. A possible reason is that these complicated methods are more prone to overfit to small datasets, or they require more hyper-parameter tuning to obtain better performance. By contrast,

the simple SiamFC has better generalization ability in more challenging and diverse scenarios.

An important observation is that, all top 18 trackers leverage deep features for tracking, which demonstrates the advantages of deep representation in achieving robust tracking performance. Moreover, we observe that model update is beneficial for achieving robust tracking, reflected by superior performance of trackers with online update (*e.g.*, DiMP and LTMU) than those without model update (*e.g.*, GlobalTrack, SiamRPN++ and SiamMask).

4.3.2 Attribute-based Performance

In order to further analyze the performance of different trackers, we conduct attribute-based evaluation.

Figure 9 shows the attribute-based evaluation results of 48 tracking algorithms with SUC scores under the full overlap protocol. From Fig. 9, we observe DiMP achieves the best performance under 13 out of 14 attributes. LTMU obtains the second best results under 11 out of 14 attributes. It is worth noting that although the three trackers LTMU, GlobalTrack, and DaSiamRPN utilize additional re-detection strategy for long-term tracking, DiMP still outperforms them under the challenge of occlusion. There are two potential reasons: First, DiMP uses a relatively larger search region for target localization. This way, DiMP can re-locate the target when it reappears. Second, DiMP adopts a more discriminative approach to update the appearance model. Thus, it shows more robust performance when the target re-appears. An interesting observation on out-of-view is that GlobalTrack and LTMU outperform DiMP, which suggests that the full image search strategy is beneficial to handle out-of-view. ATOM obtains promising performance on all attributes owing to the effectiveness of scale estimation networks. In addition, other trackers such as SiamRPN++ and SiamMask achieve competitive results on these 14 attributes. We note that all the top seven trackers, including DiMP, LTMU, GlobalTrack, DaSiamRPN, ATOM, SiamRPN++ and SiamMask, employ deeper feature representation (*e.g.*, ResNet-18 or ResNet-50 He et al. 2016) for appearance modeling, which shows the importance of powerful features for visual tracking.

4.3.3 Qualitative Evaluation

To qualitatively analyze different trackers and provide guidance for future research, we show sampled tracking results of eight top performers, including DiMP, LTMU, GlobalTrack, DaSiamRPN, ATOM, SiamRPN++, SiamMask and C-RPN, under challenges such as *fast motion*, *full occlusion*, *low resolution*, *out-of-view*, *aspect ratio change* and *background clutter* in Fig. 10.

From Fig. 10, we observe that, for sequence *yoyo-7* with *fast motion*, trackers are prone to lose the target because most current algorithms perform target localization from a relatively small region. Although DiMP, LTMU, GlobalTrack, and DaSiamRPN utilize a large search region or adopt re-detection strategies, they still fail as *fast motion* easily causes *motion blur*, which significantly affects re-localization performance of these four trackers. A possible solution to handle this issue is to combine rich temporal and motion cues with appearance information for tracking. In video *goldfish-7* with *full occlusion*, trackers drift to the background region. In order to deal with occlusion, an additional detection component is required to improve performance. All tracking algorithms fail on the video *pool-3* because of the ineffective representation for small target objects. To deal with this, one feasible strategy for deep trackers is to combine multi-scale features from various layers to incorporate details into representation. Video *basketball-11* is difficult due to the *out-of-view* challenge. Similar to the solution for handling *occlusion*, one can leverage an extra instance-level detector to re-locate the target object. *Aspect ratio change* is challenging in *train-1* as most existing trackers often adopt a simple method (*e.g.*, random search or pyramid strategy) to deal with it. A few algorithms such as ATOM and SiamRPN++ borrow techniques from detection for tracking and show promising results. However, since targets may also have *rotation* at the same time, these trackers cannot accurately localize the objects. To effectively estimate target scale, a solution is to take rotation factor into consideration. For video *person-5* with heavy *background clutter*, all trackers drift due to less discriminative representation for target and background. A possible solution to alleviate this issue is to utilize the contextual information to enhance the discriminability or fine-grained feature presentation to improve target recognition ability.

4.4 Evaluation with One-Shot Protocol

4.4.1 Overall Performance

Different from full overlap protocol, videos for evaluation in the one-shot protocol are from *unseen* categories. In LaSOT, 150 sequences (about 380 K frames) from 15 classes are used for performance assessment, and none of the 15 classes is included in the training set or in ImageNet. Figure 11 demonstrates the evaluation results of all algorithms in OPE setting. From Fig. 11, LTMU obtains the best results with a 0.473 PRE score, 0.499 N-PRE score and 0.414 SUC score. DiMP exhibits the second best performance with a 0.451 PRE score, 0.476 N-PRE score and 0.392 SUC score. ATOM achieves the third best results with PRE score of 0.43, N-PRE score of 0.459 and SUC score of 0.376. DiMP performs more robustly than ATOM because it exploits more background informa-

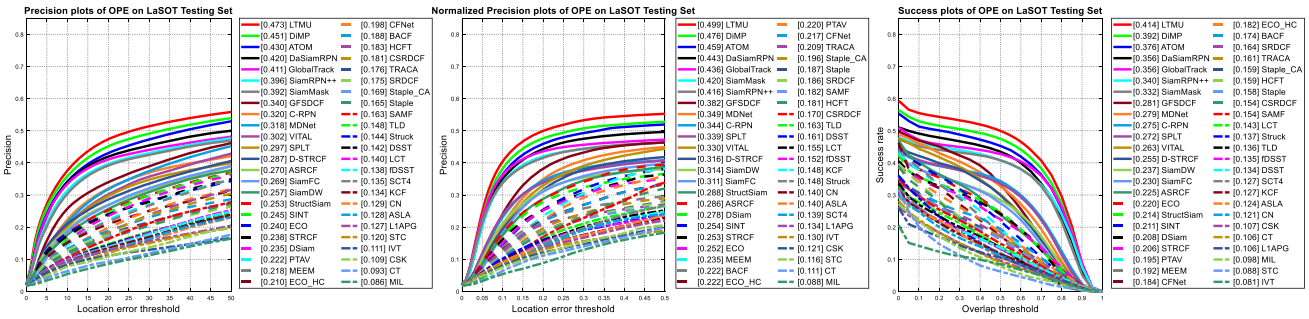


Fig. 11 Overall evaluation results on LaSOT under *one-shot* protocol. Best viewed in color

tion to improve discriminability. We also notice that, in this protocol, ATOM shows better results than DaSiamRPN with a 0.42 PRE score, 0.443 N-PRE score and 0.356 SUC score. This is may be due to their different model update strategies. GlobalTrack obtains promising performance with 0.411 PRE score, 0.436 N-PRE score and 0.356 SUC score. For targets from *unseen* categories, compared to linear template update, supervised online learning can better adapt to appearance changes during tracking, leading to more robust performance. Note that, despite the risk of drifting, online learning trackers usually select reliable tracking results based on their confidences for update. This way, the drift problem can be alleviated to some extent during updating. Similar results have shown that trackers using deeper features such as LTMU, DiMP, ATOM, DaSiamRPN, GlobalTrack, SiamRPN++ and SiamMask achieve better results.

4.4.2 Attribute-based Performance

Figure 12 demonstrates the attribute-based evaluation results of 45 trackers. We observe that DiMP achieves the best results on 10 out of 14 attributes. DiMP shows the best performance on 3 attributes and the second best results on 8 attributes, demonstrating slightly better performance than GlobalTrack, ATOM and DaSiamRPN. A surprising finding is that despite better overall results of DiMP, GlobalTrack outperforms it on the challenge of out-of-view, thanks to the global search strategy. In addition, an interesting observation is that, SiamMask, which integrates segmentation into tracking for improvement, does not show better performance than DaSiamRPN and SiamRPN++. We conjecture that it is caused by the lack of mask annotation for training SiamMask on our benchmark.

4.4.3 Qualitative Evaluation

We show qualitative results of eight trackers, including LTMU, DiMP, ATOM, DaSiamRPN, GlobalTrack, Siam RPN++, SiamMask and GFSDCT, in six representative challenges such as *fast motion*, *full occlusion*, *low resolution*, *rotation*, *background* and *scale variation* in Fig. 13. For videos with *fast motion* and *full occlusion* (*e.g.*, *badminton-1* and *cosplay-8*), trackers easily drift because they usually utilize a relatively small search for target localization. A solution is to enlarge the search region accordingly or even perform tracking on the full image. For sequences with *low-resolution* and *rotation* (*e.g.*, *frisbee-2* and *jianzi-4*), the tracking algorithms may lose the target because of ineffective feature extraction for target appearance. A feasible method to handle this issue is to mine for motion features in videos. When *background clutter* happens with many distractors (*e.g.*, *misc-10*), it is hard for trackers to locate the target. To solve this issue, one can exploit more spatial details of target to improve discriminative ability of tracking models. In addition, trackers are prone to drift when heavy *scale variation* occurs with other challenges such as *aspect ratio change* (*e.g.*, *paddle-6*). One can leverage techniques such as instance segmentation to improve scale estimation.

4.5 Retraining on LaSOT

In order to show the advantages of large-scale training set, we retrain two representative trackers SiamFC Bertinetto et al. (2016b) and CFNet Valmadre et al. (2017) using sequences from LaSOT instead of VID for video object detection. Notice that all training settings are kept the same as those for training on VID. After re-training, we compare the performance of these two trackers on different benchmarks including OTB-13, OTB-15, and $\text{LaSOT}_{\text{tst}}$ in both protocols.

Table 6 demonstrates the results of retraining using our dedicated benchmark and comparisons with the performance of the original SiamFC and CFNet trained on ImageNet VID Russakovsky et al. (2015). We observe that for both trackers, the performance is improved. Specifically on OTB-13, the SUC score of SiamFC is improved from 0.588 to 0.608 using training split in our full overlap protocol. Furthermore, because of there being more data in the one-shot protocol, the SUC score is increased to 0.614 with significant gains of 2.6%. On OTB-15, the SUC score of SiamFC is improved from 0.565 to 0.582 and 0.589 with training data from two protocol settings, respectively. Similarly, the SUC score of

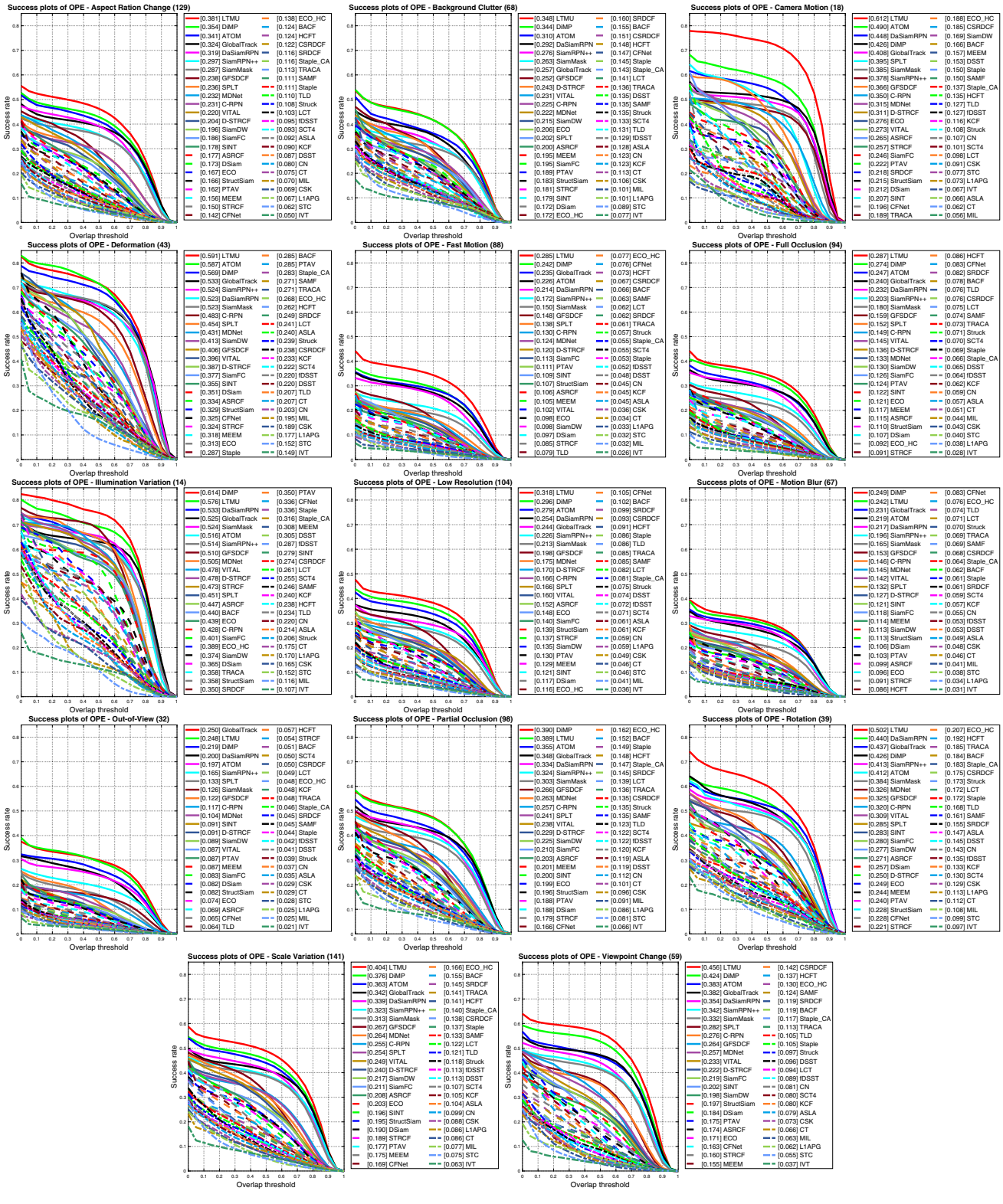


Fig. 12 Performance of trackers on each attribute using success under *one-shot* protocol. Best viewed in color

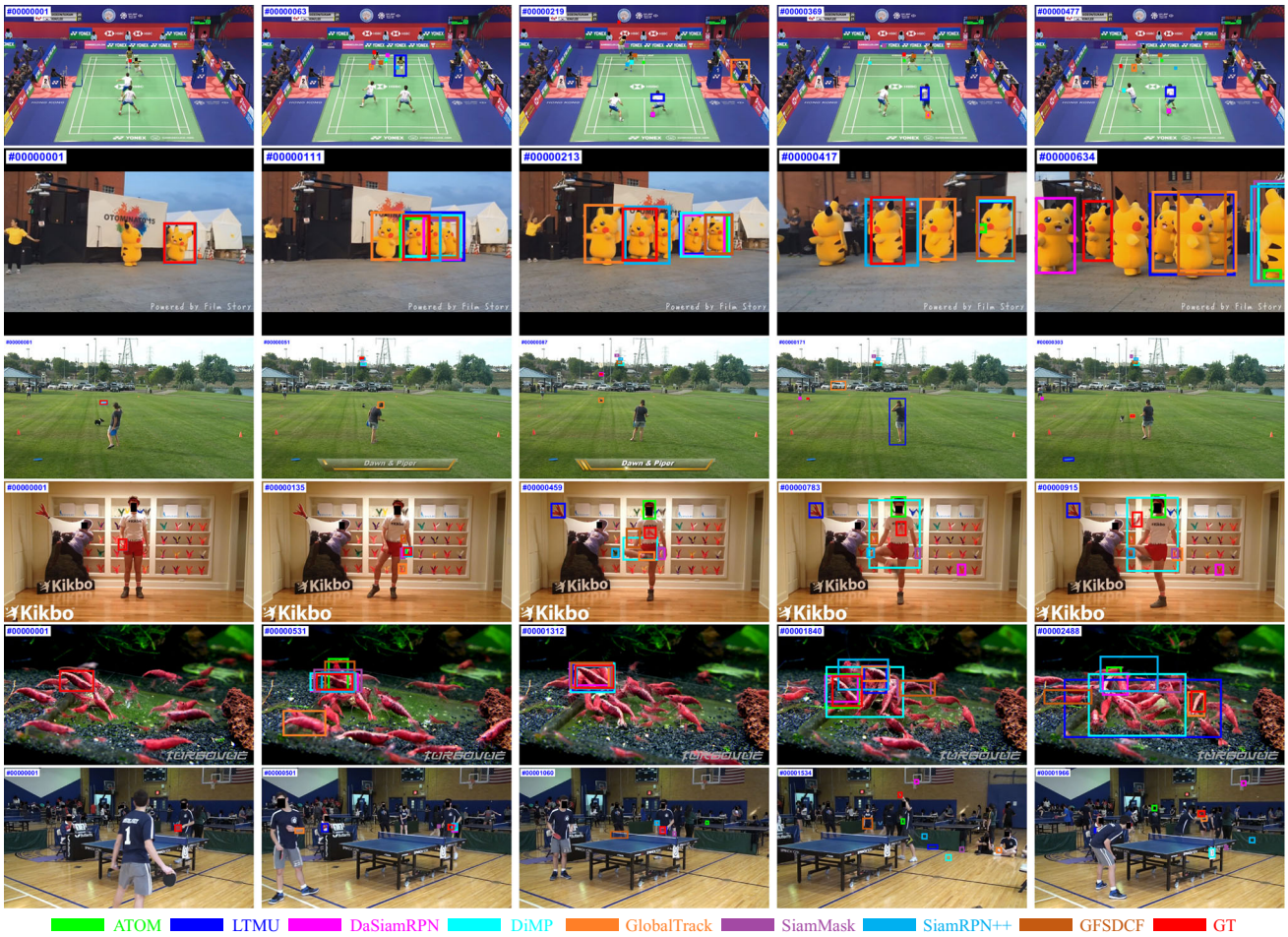


Fig. 13 Qualitative evaluation in six difficult challenging videos in the one-shot protocol (from top to bottom): *badminton-1* with *fast motion*, *cosplay-8* with *full occlusion*, *frisbee-2* with *low resolution*, *jianzi-4* with *rotation*, *misc-10* with *background clutter* and *paddle-6* with *scale variation*. Best viewed in color

CFNet obtains obvious improvements on both OTB-13 and OTB-15. More specifically, the SUC score is improved from 0.589 to 0.615 and 0.622 on OTB-13 using training sets in two protocols, and on OTB-15 the score is increased from 0.568 to 0.593 and 0.598.

In addition, we re-evaluate these two trackers on LaSOT under two protocols after retraining, as shown in Table 6. For SiamFC, the SUC scores under two different protocols are improved from 0.336 to 0.342 and from 0.230 to 0.237, respectively. For CFNet, the SUC scores are improved from 0.275 to 0.286 and from 0.184 to 0.194, respectively. These performance gains show the advantages of large-scale training dataset for improving tracking performance. It is worth noting that, the improvements on the smaller datasets OTB-13 and OTB-15 are higher than those on our testing sets. One possible reason is that our testing sequences are more challenging with focus on long-term tracking, while the original trackers are designed for short-term tracking. For better performance, one may need to adjust more hyperparameters or even design new frameworks.

5 Discussion

5.1 Full Overlap Versus One-shot

By definition, the full overlap protocol allows overlap of object classes between training and testing sets, while one-shot protocol does not allow such overlap. Not surprisingly, the one-shot protocol is more challenging because the tracking algorithms need to generalize to objects with unfamiliar appearance and motion pattern.

By comparing the success score of each tracker on the one-shot protocol against the full overlap one, we observe an obvious performance drop (by 0.037–0.18) for all algorithms. Such degradation clearly suggests that existing trackers do not fully address the domain gap between different object

Table 6 Retraining experiments of two trackers SiamFC Bertinetto et al. (2016b) and CFNet Valmadre et al. (2017) on LaSOT

Training set	SiamFC Bertinetto et al. (2016b)			CFNet Valmadre et al. (2017)		
	ImageNet VID Russakovsky et al. (2015)	$\text{LaSOT}_{\text{tra}}$ (full overlap)	$\text{LaSOT}_{\text{tra}}$ (one-shot)	ImageNet VID Russakovsky et al. (2015)	$\text{LaSOT}_{\text{tra}}$ (full overlap)	$\text{LaSOT}_{\text{tra}}$ (one-shot)
OTB-13 Wu et al. (2013)	0.588	0.608 (↑**0.020**)	0.614 (↑**0.026**)	0.589	0.615 (↑**0.026**)	0.622 (↑**0.033**)
OTB-15 Wu et al. (2015)	0.565	0.582 (↑**0.017**)	0.589 (↑**0.024**)	0.568	0.593 (↑**0.025**)	0.598 (↑**0.030**)
$\text{LaSOT}_{\text{tst}}$ (full overlap)	0.336	0.342 (↑**0.006**)	–	0.275	0.286 (↑**0.011**)	–
$\text{LaSOT}_{\text{tst}}$ (one-shot)	0.230	–	0.237 (↑**0.007**)	0.184	–	0.194 (↑**0.010**)

categories. To mitigate the performance degradation caused by such domain gap, a potential future direction is to explore domain adaption Ganin and Lempitsky (2015) for tracking by treating each category or even each target as an individual domain. In addition, by comparing all trackers within full overlap or one-shot protocols, we see that all top five trackers (see Figs. 8 and 11) employ deep features for target appearance representation, which shows that designing more effective feature representations should be paid attention to in both scenarios. Considering the dynamic nature of tracking problems, future research can leverage both spatial appearance information and motion features to improve tracking for both seen and unseen object categories. Moreover, we observe that for the top five trackers in each protocol, the best three update the model during tracking, which suggests model updating is critical for both protocols.

5.2 Short-term and Long-term Tracking Algorithms

One goal of our benchmark is to advance the development of long-term tracking algorithms. In full overlap evaluation, DiMP achieves the best results and outperforms the long-term tracker LTMU. We argue that the reasons are two-fold. First, DiMP utilizes a relatively large search region for tracking, which effectively handles the problems of full occlusion and out-of-view. Second, the update method in DiMP leverages more historic information than LTMU. In addition, long-term tracker GlobalTrack outperforms ATOM and SiamRPN++ owing to deeper feature representation and a better mechanism to locate target objects using the full image. On the other hand, in one-shot evaluation, LTMU achieves the best performance with SUC score of 0.414. Compared to LTMU, DiMP still achieves competitive results with 0.392 SUC score. The reason that LTMU outperforms DiMP in the one-shot protocol is because there are many small targets. As a result, the tracking model may fail due to ineffective feature extraction and fast target motion. LTMU employs a global search strategy to re-locate the target when drift happens, leading to better results. Moreover, we note that although GlobalTrack adopts full image search methods, its result with 0.356 is inferior in comparison to DiMP, which suggests the importance of effective model updating for robust performance.

Based on the above analysis, we argue that there are several directions that can be taken to improve long-term tracking. First, a deeper feature representation (*e.g.*, ResNet-50) can help to effectively distinguish targets from their backgrounds. Second, a larger search region may be helpful for occluded and out-of-view targets. Third, although matching based trackers (*e.g.*, GlobalTrack and SiamRPN++) achieve promising results in long-term tracking, model updating is still crucial to obtaining more robust performance (*e.g.*, LTMU and DiMP).

5.3 Analysis on Deeper Feature Representation for Tracking

Feature representation has been one of the most important components for robust tracking. In this subsection, we conduct experiments by comparing different backbones in both protocols. We choose SiamRPN++ and DiMP for experiments since both approaches provide official implementations with different backbone architectures. Specifically, we study SiamRPN++ with AlexNet, ResNet-18 and ResNet-50 and DiMP with ResNet-18 and ResNet-50. The experimental results are demonstrated in Table 7.

From Table 7, we can see that on full overlap evaluation, SiamRPN++ with AlexNet achieves a success score of 0.433 and the performance is further improved to 0.472 and 0.495 success scores using deeper architectures ResNet-18 and ResNet-50, respectively. Similarly, DiMP with deeper architecture ResNet-50 shows a better success score of 0.560, outperforming DiMP with ResNet-18 achieving 0.534 success score. Likewise, on one-shot evaluation, SiamRPN++ with deeper ResNet-50 achieves the better performance with a success score of 0.340 compared to the scores of 0.316 and 0.245 achieved with ResNet-18 and AlexNet. DiMP with ResNet-50 obtains a higher score of 0.392 than the score of 0.381 achieved with ResNet-18. The above comparison clearly suggests that feature representation learned by

Table 7 Comparison experiments of different architectures on two protocols using success score

	Architectures	Full overlap	One-shot
SiamRPN++	AlexNet	0.433	0.245
	ResNet-18	0.472	0.316
	ResNet-50	0.495	0.340
DiMP	ResNet-18	0.534	0.381
	ResNet-50	0.560	0.392

deeper networks demonstrates better robustness for tracking in both full overlap and one-shot protocols. In addition, an interesting observation is that deeper networks are crucial when dealing with unseen targets. When changing backbones from ResNet-18 to AlexNet, the performance drop for SiamRPN++ is 0.039 on the full overlap evaluation. However, on one-shot evaluation, the performance degradation is more obvious with a drop of 0.071 when using AlexNet, which shows that deeper feature representation is more important for tracking performance in locating unseen targets.

5.4 Analysis on Model Update for Tracking

Visual tacking is an ill-posed problem in which only information from the first frame is reliable. Due to target appearance variation in video, tracking models usually need an update strategy to handle appearance variation. However, because of occlusion and inaccurate intermediate results, model updating is an extremely complex process. For example, it is difficult to determine when and how to utilize current information for updates. Inappropriate updates may increase the risk of drifting. To avoid this issue, existing trackers such as GlobalTrack, SiamRPN++, SiamMask, and C-RPN formulate tracking as a matching problem without model updates. These approaches show promising performance by achieving success scores of 0.517, 0.495, 0.467 and 0.455 on full overlap evaluation and 0.356, 0.340, 0.332 and 0.275 on one-shot evaluation. In comparison to these trackers without updates, methods with model update including DiMP, LTMU, ATOM, and DaSiamRPN obtain better success scores of 0.560, 0.539, 0.515 and 0.499 on full overlap evaluation and 0.392, 0.414, 0.376 and 0.356 on one-shot evaluation. In addition, we observe that the evaluation of most attributes demonstrates that trackers with model update show better performance. Through the above comparison and analysis, we argue that although online learning for model update is not key to performance improvement, it is essential to perform model updates to achieve robust tracking. We hope that this analysis can inspire future research for better design of tracking algorithms.

6 Conclusion

In this paper, we introduced LaSOT, a high-quality large-scale single-object tracking benchmark containing 1550 videos with more than 3.87 million frames. To our knowledge, LaSOT is by far the *largest* tracking benchmark, in terms of precisely annotated frames. By releasing LaSOT, we expect to offer the community a dedicated platform to develop deep trackers and evaluate long-term tracking performance. In addition, we provided additional lingual specification for each sequence, aiming to encourage the exploration of lingual features to further improve performance. Moreover, for flexible performance evaluation we designed two different experimental settings: the full overlap and one-shot protocols. Extensive experiments on LaSOT by assessing 48 trackers indicate that there is still significant room for future improvement.

Acknowledgements We thank the anonymous reviewers for insightful suggestions, and Jeremy Chu for proofreading the final draft. Ling was supported partially by the Amazon AWS Machine Learning Research Award.

References

Babenko, B., Yang, M.H., & Belongie, S. (2009). Visual tracking with online multiple instance learning. In: CVPR.

Bao, C., Wu, Y., Ling, H., & Ji, H. (2012). Real time robust l1 tracker using accelerated proximal gradient approach. In: CVPR

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., & Torr, P.H. (2016). Staple: Complementary learners for real-time tracking. In: CVPR.

Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., & Torr, P.H. (2016). Fully-convolutional siamese networks for object tracking. In: ECCVW

Bhat, G., Danelljan, M., Gool, L.V., Timofte, R. (2019) Learning discriminative model prediction for tracking. In: ICCV

Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M. (2010). Visual object tracking using adaptive correlation filters. In: CVPR.

Choi, J., Chang, H.J., Fischer, T., Yun, S., Lee, K., Jeong, J., Demiris, Y., Choi, J.Y. (2018). Context-aware deep feature compression for high-speed visual tracking. In: CVPR

Choi, J., Jin Chang, H., Jeong, J., Demiris, Y., Young Choi, J. (2016). Visual tracking using attention-modulated disintegration and integration. In: CVPR.

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.

Dai, K., Wang, D., Lu, H., Sun, C., Li, J. (2019). Visual tracking via adaptive spatially-regularized correlation filters. In: CVPR

Dai, K., Zhang, Y., Wang, D., Li, J., Lu, H., Yang, X. (2020). High-performance long-term tracking with meta-updater. In: CVPR.

Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M. (2017). Eco: Efficient convolution operators for tracking. In: CVPR

Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M. (2019). Atom: Accurate tracking by overlap maximization. In: CVPR

Danelljan, M., Häger, G., Khan, F., Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In: BMVC.

Danelljan, M., Häger, G., Khan, F. S., & Felsberg, M. (2017). Discriminative scale space tracking. *TPAMI*, *39*(8), 1561–1575.

Danelljan, M., Hager, G., Shahbaz Khan, F., & Felsberg, M. (2015). Learning spatially regularized correlation filters for visual tracking. In: ICCV.

Danelljan, M., Robinson, A., Khan, F.S., & Felsberg, M. (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: ECCV.

Danelljan, M., Shahbaz Khan, F., Felsberg, M., Van de Weijer, J. (2014). Adaptive color attributes for real-time visual tracking. In: CVPR.

Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D. (2020). Tao: A large-scale benchmark for tracking any object. In: ECCV.

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

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*, *88*(2), 303–338.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H. (2019). Lasot: A high-quality benchmark for large-scale single object tracking. In: CVPR.

Fan, H., Ling, H. (2017). Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In: ICCV.

Fan, H., Ling, H. (2017). Sanet: Structure-aware network for visual tracking. In: CVPRW.

Fan, H., Ling, H. (2019). Siamese cascaded region proposal networks for real-time visual tracking. In: CVPR

Fan, H., Yang, F., Chu, P., Yuan, L., & Ling, H. (2020). TracKlinic: Diagnosis of challenge factors in visual tracking. In: arXiv:1911.07959.

Feng, Q., Ablavsky, V., Bai, Q., Li, G., & Sclaroff, S. (2020). Real-time visual object tracking with natural language description. In: WACV.

Galoogahi, H.K., Fagg, A., Huang, C., Ramanan, D., & Lucey, S. (2017). Need for speed: A benchmark for higher frame rate object tracking. In: ICCV.

Galoogahi, H.K., Fagg, A., Lucey, S. (2017). Learning background-aware correlation filters for visual tracking. In: ICCV.

Ganin, Y., Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In: ICML.

Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., & Wang, S. (2017). Learning dynamic siamese network for visual object tracking. In: ICCV.

Gupta, A., Dollar, P., & Girshick, R. (2019). Lvis: A dataset for large vocabulary instance segmentation. In: CVPR.

Hare, S., Saffari, A., Torr, P.H.S. (2011). Struck: Structured output tracking with kernels. In: ICCV.

He, A., Luo, C., Tian, X., Zeng, W. (2018). A twofold siamese network for real-time object tracking. In: CVPR.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: CVPR.

Henriques, J.F., Caseiro, R., Martins, P., & Batista, J. (2012). Exploiting the circulant structure of tracking-by-detection with kernels. In: ECCV.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *TPAMI*, *37*(3), 583–596.

Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., & Darrell, T. (2016). Natural language object retrieval. In: CVPR.

Huang, L., Zhao, X., & Huang, K. (2019). Got-10k: A large high-diversity benchmark for generic object tracking in the wild. TPAMI.

Huang, L., Zhao, X., & Huang, K. (2020). Globaltrack: A simple and strong baseline for long-term tracking. In: AAAI.

Jia, X., Lu, H., & Yang, M.H. (2012). Visual tracking via adaptive structural local sparse appearance model. In: CVPR.

Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-learning-detection. *TPAMI*, *34*(7), 1409–1422.

Kristan, M., Matas, J., Leonardis, A., Vojíř, T., Pflugfelder, R., Fernandez, G., et al. (2016). A novel performance evaluation methodology for single-target trackers. *TPAMI*, *38*(11), 2137–2155.

Kristan et al., M. (2017). The visual object tracking vot2017 challenge results. In: ICCVW.

Kristan et al., M. (2018). The visual object tracking vot2018 challenge results. In: ECCVW.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In: NIPS.

Li, A., Lin, M., Wu, Y., Yang, M. H., & Yan, S. (2016). Nus-pro: A new visual tracking challenge. *TPAMI*, *38*(2), 335–349.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., & Yan, J. (2019). Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: CVPR.

Li, B., Yan, J., Wu, W., Zhu, Z., & Hu, X. (2018). High performance visual tracking with siamese region proposal network. In: CVPR.

Li, F., Tian, C., Zuo, W., Zhang, L., Yang, M.H. (2018). Learning spatial-temporal regularized correlation filters for visual tracking. In: CVPR.

Li, P., Chen, B., Ouyang, W., Wang, D., Yang, X., & Lu, H. (2019). Gradnet: Gradient-guided network for visual object tracking. In: ICCV.

Li, P., Wang, D., Wang, L., & Lu, H. (2018). Deep visual tracking: Review and experimental comparison. *Pattern Recog.*, *76*, 323–338.

Li, S., Xiao, T., Li, H., Zhou, B., Yue, D., & Wang, X. (2017). Person search with natural language description. In: CVPR.

Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., & Hengel, A. V. D. (2013). A survey of appearance models in visual object tracking. *ACM TIST*, *4*(4), 58.

Li, Y., & Zhu, J. (2014). A scale adaptive kernel correlation filter tracker with feature integration. In: ECCVW.

Li, Z., Tao, R., Gavves, E., Snoek, C.G., & Smeulders, A.W., et al. (2017). Tracking by natural language specification. In: CVPR.

Liang, P., Blasch, E., & Ling, H. (2015). Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, *24*(12), 5630–5644.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014) Microsoft coco: Common objects in context. In: ECCV.

Liu, T., Wang, G., & Yang, Q. (2015) Real-time part-based visual tracking via adaptive correlation filters. In: CVPR

Lukezic, A., Kart, U., Kapyla, J., Durmush, A., Kamarainen, J.K., Matas, J., Kristan, M. (2019). Cdtb: A color and depth visual object tracking dataset and benchmark. In: ICCV.

Lukezic, A., Vojir, T., Zajc, L.C., Matas, J., & Kristan, M. (2017). Discriminative correlation filter with channel and spatial reliability. In: CVPR.

Ma, C., Huang, J.B., Yang, X., & Yang, M.H. (2015) Hierarchical convolutional features for visual tracking. In: ICCV

Ma, C., Yang, X., Zhang, C., & Yang, M.H. (2015). Long-term correlation tracking. In: CVPR.

Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831.
Mueller, M., Smith, N., & Ghanem, B. (2016). A benchmark and simulator for uav tracking. In: ECCV.
Mueller, M., Smith, N., & Ghanem, B. (2017). Context-aware correlation filter tracking. In: CVPR.
Müller, M., Bibi, A., Giancola, S., Al-Subaihi, S., & Ghanem, B. (2018). Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV
Nam, H., Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In: CVPR.
Real, E., Shlens, J., Mazzocchi, S., Pan, X., & Vanhoucke, V. (2017) Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In: CVPR
Ross, D. A., Lim, J., Lin, R. S., & Yang, M. H. (2008). Incremental learning for robust visual tracking. *IJCV*, *77*(1–3), 125–141.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *IJCV*, *115*(3), 211–252.
Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In: ICLR.
Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2014). Visual tracking: An experimental survey. *TPAMI*, *36*(7), 1442–1468.
Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C., Lau, R., & Yang, M.H. (2018). Vital: Visual tracking via adversarial learning. In: CVPR.
Tao, R., Gavves, E., & Smeulders, A.W. (2016). Siamese instance search for tracking. In: CVPR.
Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H. (2017). End-to-end representation learning for correlation filter based tracking. In: CVPR.
Valmadre, J., Bertinetto, L., Henriques, J.F., Tao, R., Vedaldi, A., Smeulders, A., Torr, P., & Gavves, E. (2018). Long-term tracking in the wild: A benchmark. In: ECCV.
Wang, G., Luo, C., Xiong, Z., & Zeng, W. (2019) Spm-tracker: Series-parallel matching for real-time visual object tracking. In: CVPR.
Wang, L., Ouyang, W., Wang, X., Lu, H. (2015). Visual tracking with fully convolutional networks. In: ICCV.
Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., & Li, H. (2019). Unsupervised deep tracking. In: CVPR.
Wang, N., & Yeung, D.Y. (2013). Learning a deep compact image representation for visual tracking. In: NIPS.
Wang, Q., Zhang, L., Bertinetto, L., Hu, W., & Torr, P.H. (2019). Fast online object tracking and segmentation: A unifying approach. In: CVPR.
Wu, Y., Lim, J., & Yang, M.H. (2013). Online object tracking: A benchmark. In: CVPR.
Wu, Y., Lim, J., & Yang, M. H. (2015). Object tracking benchmark. *TPAMI*, *37*(9), 1834–1848.
Xu, T., Feng, Z.H., Wu, X.J., & Kittler, J. (2019). Joint group feature selection and discriminative filter learning for robust visual object tracking. In: ICCV.
Yan, B., Zhao, H., Wang, D., Lu, H., Yang, X. (2019). 'skimming-perusal'tracking: A framework for real-time and robust long-term tracking. In: ICCV.
Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM CSUR*, *38*(4), 13.
Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In: NIPS.
Zhang, J., Ma, S., & Sclaroff, S. (2014). Meem: robust tracking via multiple experts using entropy minimization. In: ECCV.
Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H. (2014). Fast visual tracking via dense spatio-temporal context learning. In: ECCV.
Zhang, K., Zhang, L., & Yang, M.H. (2012). Real-time compressive tracking. In: ECCV.
Zhang, Y., Wang, L., Qi, J., Wang, D., Feng, M., & Lu, H. (2018). Structured siamese network for real-time visual tracking. In: ECCV
Zhang, Z., & Peng, H. (2019). Deeper and wider siamese networks for real-time visual tracking. In: CVPR.
Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene parsing through ade20k dataset. In: CVPR.
Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., & Hu, W. (2018). Distractor-aware siamese networks for visual object tracking. In: ECCV.