Tiny Object Tracking: A Large-Scale Dataset and a Baseline

Yabin Zhu, Chenglong Li, Yao Liu, Xiao Wang, Member, IEEE, Jin Tang, Bin Luo, Senior Member, IEEE, and Zhixiang Huang, Senior Member, IEEE

Abstract—Tiny objects, frequently appearing in practical applications, have weak appearance and features, and receive increasing interests in many vision tasks, such as object detection and segmentation. To promote the research and development of tiny object tracking, we create a large-scale video dataset, which contains 434 sequences with a total of more than 217K frames. Each frame is carefully annotated with a high-quality bounding box. In data creation, we take 12 challenge attributes into account to cover a broad range of viewpoints and scene complexities, and annotate these attributes for facilitating the attribute-based performance analysis. To provide a strong baseline in tiny object tracking, we propose a novel multilevel knowledge distillation network (MKDNet), which pursues three-level knowledge distillations in a unified framework to effectively enhance the feature representation, discrimination, and localization abilities in tracking tiny objects. Extensive experiments are performed on the proposed dataset, and the results prove the superiority and effectiveness of MKDNet compared with state-of-the-art methods. The dataset, the algorithm code, and the evaluation code are available at https://github.com/mmic-lcl/Datasets-and-benchmark-code.

Index Terms—Benchmark dataset, knowledge distillation, tiny objects, visual tracking.
network. Since the teacher network sometimes has wrong
as the supervision to guide the learning of the student
distillation scheme that uses the IoU score of the teacher
ability of the student network, we also design an IoU-level
learning of the student network. To improve the localization
from the teacher network as the supervision to guide the
aims to use the potential distribution of classification output
from tiny objects. To solve this problem, we propose a
novel multilevel knowledge distillation network (MKDNet) to
effectively enhance the feature representation, discrimi-
nation, and localization abilities in tiny object tracking.
4) Extensive experiments on the proposed dataset validate
the superiority and effectiveness of our MKDNet.
We also report the results of 24 state-of-the-art trackers
on the proposed dataset for the evaluation of different
tracking algorithms. We believe that our dataset and
baseline would promote the research and development
of tiny object tracking.

II. RELATED WORK

In this section, we give a brief introduction to visual
tracking, small object perception, and knowledge distilla-
tion. More related works can be found in the following
surveys [22], [23], [24].

A. Visual Tracking

In recent years, a large number of trackers [20], [21],
[25], [26], [27], [28], [29], [30], [31], [32] have been
proposed to solve various challenges in visual tracking. The
first category is trackers based on multidomain learning,
including multidomain network (MDnet) [20] and real-
time MDNet (RT-MDNet) [25]. MDNet and RT-MDNet
distinguish foreground objects from background online by
using the binary classifier. The second category is Siame-
based trackers, including SiamRPN [28], SiamRPN++ [29],
GlobalTrack [30], and Siam R-CNN [31]. SiamRPN and
SiamRPN++ add the region proposal network module which
improves not only the tracking performance but also the
speed. GlobalTrack and Siam R-CNN aim to solve long-term
tracking challenges by using a wide range of search strategies.
The third category is trackers based on discriminant feature
learning, including accurate tracking by overlap maximization
(ATOM) [32] and learning discriminative model prediction
(DiMP) [21]. ATOM and DiMP are the combination of
Siamese network and correlation filtering model in tracking,
and they introduce the scale regression method [33] into
the tracking framework. However, these methods do not focus on
the problem of tiny objects, which have many challenges, such
as low resolution, image blur, less effective information, and
more noises. Therefore, it is difficult for general trackers to
track tiny objects accurately.
B. Small Object Perception

The perception of small objects lies in multiple tasks such as object detection [4], [5], semantic segmentation [6], and object tracking [34]. In object detection, Noh et al. [4] propose a tiny object feature enhancement strategy based on generative adversarial network (GAN), which effectively alleviates the problem of weak features of small objects. Kisantal et al. [5] propose to boost the detection accuracy of small objects by solving the problem of an unbalanced distribution of small objects. In semantic segmentation, Guo et al. [6] propose a measurement method to effectively increases the contribution of small objects to the overall loss, thereby improving the segmentation accuracy of small target instances. However, these methods can not directly solve the problem of tiny object tracking as it aims to track tiny objects in video sequences instead of images. In visual tracking, Marvasti-Zadeh et al. [34] combine the context-awareness strategy and the multiscale feature aggregation method to solve the object tracking problem under aerial photography. But it is difficult to solve the problem of tiny objects in natural scenes. But it only aggregates multiscale features and does not substantially enhance the quality of tiny object. Therefore, it is difficult to solve the problem of tiny objects tracking.

C. Knowledge Distillation

A lot of works [35], [36], [37], [38], [39], [40] use knowledge distillation to enhance the performance of convolutional neural networks. These methods of knowledge distillation can be roughly divided into three categories, including logits distillation, feature distillation, and regression distillation. The logits distillation methods [35], [36] mainly enhance the performance of the student network by restricting the classification distribution of the student network to be consistent with the teacher network. The feature distillation methods [37], [38] are based on original knowledge distillation. Their core idea is to use not only the output of the teacher network but also the features of the intermediate layer of the teacher network as the knowledge to boost the training of the student network. The regression distillation methods [39], [40] explore the application of knowledge distillation in the regression model. Inspired by these methods, we propose an MKDNet to deal with the problem of tiny object tracking.

III. LaTOT Dataset

In this section, we introduce the details of our LaTOT benchmark dataset, including the definition of tiny object in video, data collection, data annotation, statistics, and analysis.

A. Definition of Tiny Object in Video

First of all, it should be pointed out that there is no definition of tiny object in video sequences. Several definitions of tiny object are for the image. In our case, we need to define tiny object in video sequences rather than in images. To this end, we refer to [48] and [49] to define tiny object in video sequences. In specific, we take the ratio of the area of the object bounding box of each frame to the area of the input image as the relative size; then average the relative sizes of all frames in each video sequence to obtain the average relative size. In this work, we set the threshold of average relative size as 1%, which is similar to existing work [48].

However, the average relative size may mistake some large object sequences as tiny object sequences. For example, in a video sequence, the average size of the object bounding box is 100 × 100, and the input image size is 1000 × 1000. According to the average relative size, it is still considered as a tiny object sequence, which is obviously unreasonable. Therefore, we also consider the average absolute size. Referring to [49], we set the threshold of average absolute size to 22 × 22 pixels. We determine whether a video is tiny or not by judging that both the average absolute size and the average relative size are below the predefined thresholds.

B. Data Collection and Annotation

Most sequences of LaTOT are collected from public video platforms, including TikTok, Bilibili, Tencent Video, and Xigua Video. There is also a small part of the dataset captured by Huawei nova 7 and IPhone 11. LaTOT contains a total of 434 video sequences with averaging frames of 501. From Table I, the numbers of tiny object sequences, categories, and total frames of our dataset are significantly larger than existing popular tracking benchmarks.

Dense and ultraprecision annotation is essential for the fair evaluation of trackers. To this end, we manually annotate each video frame in LaTOT and perform multistage careful inspections and modifications. To accomplish the labeling work, we set up an annotation team, which includes a Ph.D. student and 15 masters working in the visual tracking field. Moreover, we choose three team members as the quality assessors. If the annotation results are not unanimously agreed by the quality assessors, they will be sent back to the annotation team for modification.

As shown in Fig. 1, some target objects are very small and often encounter other challenges such as fast motion, motion blur, low resolution, and similar object. It usually results in nearly 60% of the dataset as unqualified labeled ones in the first round of annotation. To handle this problem, we enlarge each frame for reannotation to ensure ultraprecision annotations. Moreover, our labeling accuracy can reach two decimal places. After three stages of inspection and labeling by assessors and labelers, the labeling work is painstakingly completed. We do our best to ensure that this dataset has high-quality dense annotations. A demo video of our proposed LaTOT benchmark dataset can be found on this link.1

1https://www.youtube.com/watch?v=IYYTLAOsa-E
C. Statistics and Analysis

1) Object Category: The diversity of object categories helps to test the performance of trackers on different categories of tiny objects. In LaTOT, we elaborately collect 48 categories for tiny objects from 270 scenes, and the distribution of video numbers is reported in Fig. 2. We notice that the data distribution of LaTOT on object categories conforms to the long-tail distribution, in which the learning under this unbalanced data is an important topic in practical applications. It can encourage the exploration of more practical and extensible tiny object tracking method.

2) Challenge Attributes: To test the performance of different tracking algorithms on various challenge attributes, we define 12 kinds of challenges as shown in Table II. We also show the distribution of each attribute in our dataset in Fig. 3. Note that some challenges defined in other tracking datasets are removed, including deformation, in-plane rotation, and out-of-plane rotation, because the tiny objects lack sufficient appearance information to aware these challenges.

In particular, we add the challenge attribute of fast motion (including high speed motion), which is very difficult for the existing methods to track tiny object that moves at high speed, but humans can easily track them by using the historical trajectory information of an object. Therefore, we hope that these tracking sequences can encourage researchers to explore how to make machines track tiny object moving at high speed like human beings.

Due to the unique property of tiny objects, trackers are very fragile to track such targets. Here, we give a detailed analysis

collect the video data publicly available on the web and shoot some video sequences without sensitive information.
a) Low resolution: Although low-resolution attribute is not included in the attribute table, they are presented in all video sequences. Tiny object is with low resolution and small size, and thus lack enough appearance information. Most of recent works are based on deep networks, which can extract powerful features of object. However, directly inputting tiny object regions into deep networks will cause their features to be very weak. If we enlarge them directly, it will not only cause image blur and sawtooth phenomenon, which could damage the representation of tiny object, but also increase the computational cost. Therefore, the low resolution is a critical factor in tiny object tracking.

b) Background interference: Compared with general object, tiny ones have small proportions in the whole image, which are easy to be interfered by cluttered background. In specific, by observing precision rate (PR) curves in Section V-B, the overall curves with the distance threshold greater than 5 are all relatively flat, which indicates that once the object is lost (caused by fast motion, out-of-view, occlusion and abrupt motion, etc.), it is difficult for trackers to track them again. The general redetection-based tracking algorithms [31], [50] can expand the search range and even carry out full image search to alleviate this problem. However, due to the particularity of tiny object, it is easy to be interfered by a large number of background objects and noises. Therefore, the background interference is another critical factor in tiny object tracking.

3) Dataset Splitting: It is necessary to provide a training set to train deep trackers for tiny object tracking. To this end, our LaTOT is divided into a training set and a testing set. In specific, 269 of which are divided into training set with 104,910 frames, and the rest video sequences are used as testing set with 112,780 frames. It is worth noting that we carefully select 260 video sequences with representative challenges in advance, and then randomly selected 165 of them as the testing set.

IV. MULTILEVEL KNOWLEDGE DISTILLATION NETWORK

In this section, we will present the details of the MKDNet, including the multilevel knowledge distillations and the implementation details in training and tracking phases.

A. Overview

As shown in Fig. 4, our proposed MKDNet mainly contains two subnetworks, i.e., a teacher network and a student network. Note that the two subnetworks share the same network architecture (i.e., the baseline tracker Super-DiMP). The major difference between two networks is their inputs. Specifically, the input of teacher network is the high-resolution object image, while the input of student network is the low-resolution object image degraded from high-resolution one. In the training phase, we use feature-, score-, and IoU-level distillation strategies to train the student network under the guidance of teacher network, which aims at making the student network approaches or even exceeds the teacher network. More importantly, we design a reliable distillation measure to avoid wrong and invalid distillations in every batch, which takes the loss of the teacher network as the distillation upper

| Attribute | Description |
|-----------|-------------|
| SV        | Scale Variation: the ratio of bounding box area is outside the range [0.5, 2] after 1s. |
| FM        | Fast Motion: the motion of the ground truth bounding box is larger than the size of the bounding box. |
| OV        | Out-of-View: some portion of the target leaves the camera field of view. |
| IV        | Illumination Variation: the illumination of the target changes significantly. |
| CM        | Camera Motion: abrupt motion of the camera. |
| MB        | Motion Blur: the target region is blurred due to the motion of the target or camera. |
| BC        | Background Clutter: the background near the target has a similar color or texture as the target. |
| SO        | Similar Object: there are the object of similar shape or same type near the target. |
| PO        | Partial Occlusion: the target is partially occluded. |
| FO        | Full Occlusion: the target is fully occluded. |
| AM        | Abrupt Motion: Abrupt motion of the camera or target. |
| LI        | Low Illumination: The illumination intensity in the target area is low. |
bound, to control the distillation process. In the testing phase, we only use the student network to carry out tracking.

B. Multilevel Knowledge Distillation

In this section, we will describe the details of MKDNet, including image degradation, feature-, score-, and IoU-level distillation strategies.

1) Image Degradation: Existing tracking datasets do not have paired high-resolution and low-resolution images. Therefore, we have to degrade images in existing tracking datasets to low-resolution ones for the simulation of tiny objects. First, we calculate the average size of ground truth bounding boxes of objects in each batch, and divide it by a scaling factor (we set it as 16 in this work). Then, the bicubic interpolation is used to downsample input images. Since the input image size of the network is fixed, we randomly choose nearest or bilinear interpolation to upsample images to the fixed size, and the input size is 352 × 352 pixels in this work.

2) Feature-Level Distillation: Due to the lack of sufficient appearance information, it is difficult to extract high-quality features of tiny objects. To transform low-resolution features into high-resolution ones, we design a feature-level distillation module based on generative adversarial learning. We treat the feature extractor of the network as the generator, and utilize the high-resolution features of teacher network to guide the learning of low-resolution features of tiny objects. We design a discriminator which contains three fully connected layers to distinguish the true and false labels of high-resolution features and generated features. The generator and discriminator can be optimized in an alternative manner

$$L_{gen} = - \sum_{i=1}^{N} \log D(S_f^i)$$

$$L_{dis} = - \sum_{i=1}^{N} \left( \log D(T_f^i) + \log \left(1 - D(S_f^i)\right) \right)$$

(1)

where $S_f^i$ and $T_f^i$, respectively, represent the features of student network and teacher network. $D(\cdot)$ is the discriminator. During this process, we design a feature consistency $\ell_1$ loss as

$$L_{cons} = \frac{1}{N} \sum_{i=1}^{N} \left\| S_f^i - T_f^i \right\|_1 .$$

(2)

To avoid irrelevant background interference, MKDNet uses the features of the region of interest extracted by PrPool [33] for distillation learning, instead of features of the entire image used in existing feature-level knowledge distillation networks [51]. As shown in Fig. 5, the feature representation ability of our tracker for tiny object is significantly improved against the baseline tracker.

3) Score-Level Distillation: Note that the potential distribution of the output prediction of teacher network can be a guidance for the learning of student network [35], [36]. Therefore, we design a score-level distillation scheme, which uses the classification loss of teacher as the supervision to teach the learning of student network, so as to improve the discrimination ability of student network. The loss of
where distillation loss can be expressed as supervision to guide the learning of student network. The IoU-distillation scheme that uses the IoU score of teacher as the ability of student network, we also design an IoU-level distillation upper bound, to control the distillation process.

5) Reliable Distillation Measure: Equations (2)–(4) actually require that the teacher model is better than the student network in every batch; however, it may not always hold up in the training phase. As the teacher network sometimes has wrong guidance supervision, direct usage of information from teacher network would lead to misleading learning of student network. To address this issue, we design a reliable distillation measure which takes the loss of the teacher network as the measure of regression branch of IoU and classification branch, respectively.

4) IoU-Level Distillation: To improve the localization ability of student network, we also design an IoU-level distillation scheme that uses the IoU score of teacher as the supervision to guide the learning of student network. The IoU-level distillation loss can be expressed as

\[
L_{\text{iou-d}} = \frac{1}{N} \sum_{i=1}^{N} \| S_{i}^{\text{iou}} - T_{i}^{\text{iou}} \|_1
\]

where \( S_{i}^{\text{iou}} \) and \( T_{i}^{\text{iou}} \) indicate the IoU score of student and teacher networks, respectively. The score-level and IoU-level distillation schemes can also be regarded as regularization of MKDNet which can avoid overfitting to some extent.

The learning rate decays by 0.2 every 15 epochs. In addition, we rate decay mechanism to maintain the stability of the network, and also adopt a small learning rate and learning rate decay mechanism to maintain the stability of the network.

5) Reliable Distillation Measure: Equations (2)–(4) actually require that the teacher model is better than the student network in every batch; however, it may not always hold up in the training phase. As the teacher network sometimes has wrong guidance supervision, direct usage of information from teacher network would lead to misleading learning of student network. To address this issue, we design a reliable distillation measure which takes the loss of the teacher network as the distillation upper bound, to control the distillation process. Specifically, when the loss of the teacher is greater than the loss of the student network, the signal from the teacher network will be ignored directly. The reliable distillation measure can be written as

\[
\text{RDM}^{\text{iou}} = \max(0, L_{i}^{\text{iou}} - L_{i}^{\text{rou}})
\]

\[
\text{RDM}^{\text{cls}} = \max(0, L_{i}^{\text{cls}} - L_{i}^{\text{cl}})
\]

where \( \text{RDM}^{\text{iou}} \) and \( \text{RDM}^{\text{cls}} \) are the reliable distillation measure of regression branch of IoU and classification branch, respectively.

The learning rate decay mechanism to maintain the stability of the network.

In this work, the learning rate of the classifier and bounding box regressor is 0.00005 and 0.0005, respectively. The learning rate decays by 0.2 every 15 epochs. In addition, it is worth noting that we fix the parameters of the teacher network, only train the student network. The entire training process is performed on the training set of TrackingNet [16], LaSOT [17], and GOT-10K [18].

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
2) Online Tracking: In the online tracking phase, we only use the student network, and the detailed operation of our method is exactly the same as baseline tracker Super-DiMP. First, the image regions of the reference image and search image corresponding to five times the object size are used as the network input and are uniformly resized to 352 × 352 pixels. Then we use the trained ResNet50 to extract the image region features of the reference image and search image. The block 4 is used for object classification, and both block 3 and block 4 are used for object estimation. It is worth noting that in the online inference phase, the ResNet50 parameters of each frame are shared, and no training tuning is performed.

After extracting the features of the reference image and search image, the object classification will be performed. The object classification branch contains a 3 × 3 convolutional layer w_1 and a target model predictor. w_1 is used to reduce the feature channel’s dimension of reference image and search image to 256. And the target model predictor generates the weights of the object classifier. Specifically, input the features of the reference image and the object bounding box to target model predictor. The target model predictor uses a 3 × 3 convolution layer w_2 to do feature mapping, and uses PrPool to extract the features of the object as the initial target model. Then the discriminative model on S_train and optimization strategy [32] based on conjugate gradient and Gauss–Newton are used to obtain the final target model. Finally, we use the target model and the features of the search image to perform correlation operations to obtain the classification score map of the object.

After obtaining the object classification score, the object estimation branch will be executed. Specifically, we find the 2-D-position with the maximum confidence score of target classification score. Together with the previously estimated object width and height, the initial bounding box B is generated. Then we add uniform random noise to B to generate a set of ten initial proposals. We maximize the predicted IoU of each box using ten gradient ascent iterations with a step length of 1. The final prediction is obtained by taking the mean of the three bounding boxes with highest IoU.

It needs to be explained in detail that the predicted IoU in each iteration is calculated as follows: 1) inputting the features and target bounding box of the reference image, we use a 3 × 3 convolution layer, a PrPool and a 1 × 1 convolution layer to obtain the modulation feature M_feat; 2) two 3 × 3 convolution layers are used to map the features of the search image to obtain IoU_feat; 3) in M_feat and IoU_feat, we perform correlation operation and extract the features of the modulated candidate box with PrPool; and 4) the features of the candidate box are operated in the fully connected layer.

All of the above-mentioned operations will be performed on the features of block 3 and block 4. Finally, we use a fully connected layer to predict the IoU.

In short, the above-mentioned operations are summarized as follows. Given the first frame with ground truth bounding box, the data augmentation method [21] is used to build an initial training set S_train which includes 15 samples. The target model can be learned with discriminative model, and updated every 20 frames or when an interference peak is detected. In addition, the IoU predictor [32] is used to further refine the target bounding boxes.

V. EXPERIMENT

In this section, we conduct extensive experiments on our newly proposed LaTOT benchmark dataset. Specifically, we will first introduce the experimental settings, including implementation details, evaluation protocols, metrics, and baseline trackers. Then, we report our results and compare with other trackers in Section V-B. After that, we analyze the tracking results under each challenging attributes in Section V-C. We also give extensive ablation studies, analysis on training data, qualitative evaluation, and failed cases, in Sections V-D–V-G, respectively.

A. Evaluation Setting

1) Implementation Details: The experiments are conducted on two platforms with I7-9800k CPU, NVIDIA RTX 2080Ti GPU and I9-9900K CPU, NVIDIA RTX 2080Ti GPU. All the experiments of our method are implemented in Python-3.7.9 using PyTorch-1.4 and operate at about 43 frames per second (FPS) with ResNet-50 backbone. We train our network for 50 epochs, and each epoch with 20,000 videos. Each video is with three training and three testing images. We employ the ADAM optimizer [53] with learning rate delay of 0.2 every 15 epochs to train the proposed network. The learning rate of the classifier and bounding box regressor are 0.00005 and 0.0005, respectively. The layer of backbone is trained with the learning rate of 0.00005.

It should be noted that in the training phase, the proposed network needs to use the teacher network to guide the learning of student network, which introduces a large number of parameters and calculations. However, in the online tracking stage, the teacher network will be removed. Therefore, compared with the baseline method, our method does not increase any parameters and computations in online tracking stage, and the running speed of the proposed method thus reaches 43 FPS on tiny object datasets.

2) Evaluation Protocols: We use two evaluation protocols for evaluating trackers. First, all the 434 sequences are adopted to evaluate the performance of tracking algorithms. It aims to provide large-scale evaluations for tracking algorithms. Second, a testing set is used to evaluate trackers. On one hand, such setting reduces the burden of evaluation of tracking algorithms. On the other hand, it provides a training set for facilitating training deep trackers to handle the challenges of tiny objects.

3) Evaluated Algorithms: We evaluate 24 latest and representative trackers on our benchmark. These tracking algorithms cover the mainstream researches on current tracking field, i.e., Dimp50 [21], Super-DiMP, PrDimp50 [52], ATOM [32], LTMU [50], SiamCAR [54], SiamRPN++ [29], CLNet [55], ECO [56], SiamBAN [57], SiamMask [58], SiamFC++ [59], Ocean [60], SiamRPN [28], D3S [61], SiamDW [62], DaSiamRPN [63], py-MDNet [20], RT-MDNet [25], py-Vital [64], Meta-Tracker [65], SiamR-CNN [31], and GlobalTrack [30].
It is worthy to note that the Super-DiMP is an improved version of DiMP, which combines the standard DiMP classifier and PrDiMP boundary box regressor. In addition, the authors of LTU give a lot of applications of their algorithms on different baselines. We choose the strongest combination on the official homepage of LTU, namely SuperDiMP + MU.

4) Evaluation Metrics: In this work, we use PR, normalized PR (NPR), and success rate (SR) to evaluate all trackers. The PR shows the percentage of frames whose estimated locations are below the given distance threshold of the ground truths. To rank the trackers, the threshold is usually set to 20 pixels in [8]. However, the threshold of 20 is somewhat unreasonable for tiny object tracking; therefore, we set the threshold to five pixels. As the precision metric is sensitive to the target size and image resolution, we normalize the precision over the size of the ground truth bounding box to calculate the NPR. With the normalized precision metric, we use the area under the curve (AUC) between 0 and 0.5 to rank the tracking algorithm. The SR is the ratio of successful frames whose overlap of the prediction bounding box and the ground truth bounding box is greater than a predefined threshold. In this work, all trackers are ranked by using the AUC between 0 and 1.

B. Evaluation Results

We show the evaluation results of all tracking algorithms on LaTOT and LaTOT testing set in Fig. 6, respectively. From the results, we notice that the type of algorithms (DIMP50, Super-Dimp, PrDimp50, ATOM, and LTU) combining the Siamese network and IoUNet [33] network have more robust performance. They benefit from good scale regression techniques and online learning strategies. SiamCAR is an anchor-free tracker, which performs better than all anchor-based trackers. Compared with other tracking methods, the performance of the anchor-based Siamese tracking algorithms is mediocre. Compared with the trackers with large-scale training, py-MDNet, py-Vital, RT-MDNet, and Meta-tracker have poor performance, which might be because the imprecise ridge regressor limits the performance of them. GlobalTrack performs searching in whole images and completely discards the prior information of the target object’s position in the previous frames. Since the objects in our dataset are too small, it is difficult for the algorithm to effectively extract the features of the small objects, and too much background interference makes it difficult to match the target object. Therefore, the GlobalTrack performs particularly poorly on our dataset. Looking at Fig. 6, we can find that these trackers have poor performance on the testing set, which shows that the testing set is more challenging than the entire LaTOT dataset. Compared with other algorithms, our method achieves the best results. Specifically, compared with the baseline tracker and the second best tracker (DiMP-50), MKDNet improves the performance by 2.9% PR, 2.3% NPR, 1.2% SR, and 1.7% PR, 0.9% NPR, 1.6% SR on LaTOT, respectively. It proves that our multilevel knowledge distillations can effectively alleviate the challenges brought by the tiny object.

In addition, we compare the results of some representative algorithms in several mainstream datasets and show them in Table III. From Table III, we can clearly see that these algorithms have significantly low accuracy on our dataset. It suggests that our tiny object tracking dataset bring very big challenges for existing tracking algorithms.

C. Attribute-Based Results

To analyze the performance on different challenges, we evaluate 24 trackers on 12 attributes on LaTOT. The results are reported in Table IV. Compared with other algorithms, our tracker can well mitigate the challenges of background clutter, similar object, illumination variation, partial occlusion, scale variation, fast motion, and camera motion. It is important to emphasize that these challenges of background clutter, partial occlusion, and similar object will significantly affect
the quality of object features. These challenges can be handled well by our MKDNet, which proves that our tracker can significantly improve the representation, discrimination, and localization abilities for tiny object.

By further observing the performance of the challenge attributes of out-of-view, full occlusion, and abrupt motion, we can find that though our method can alleviate some challenge of tiny object tracking to a certain extent, there is still a big gap to solve the problem of tiny object tracking. Therefore, the research and development of tiny object tracking have a long way to go in real-world applications.

**D. Ablation Study**

In this section, we will analyze the impact of the main components on tracking performance. In specific, we report the tracking results of five versions of our tracker on LaTOT, as shown in Table V. They are MKDNet-FD, MKDNet-SD, MKDNet-IoUD, MKDNet-FD-GAN, and MKDNet-noRDM. MKDNet-FD, MKDNet-SD, and MKDNet-IoUD, respectively, indicate that only feature-level, score-level, and IoU-level distillation are used. MKDNet-FD-GAN indicates that we only use adversarial loss in feature-level distillation. MKDNet-noRDM indicates that reliable distillation measure is removed in the MKDNet.

| Tracker          | SV       | FM       | OV       | IV       | CM       | MB       | BC       | SO       | PO       | FO       | AM       | LI       |
|------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| MKDNet           | 0.441    | 0.264    | 0.171    | 0.474    | 0.378    | 0.141    | 0.404    | 0.448    | 0.299    | 0.227    | 0.156    | 0.522    |
| LTMU [50]        | 0.43     | 0.261    | 0.169    | 0.46     | 0.355    | 0.148    | 0.373    | 0.416    | 0.281    | 0.22     | 0.157    | 0.514    |
| Super-DiMP [52]  | 0.424    | 0.257    | 0.164    | 0.445    | 0.371    | 0.147    | 0.386    | 0.422    | 0.291    | 0.218    | 0.155    | 0.517    |
| PrDiMP50 [52]    | 0.429    | 0.26     | 0.157    | 0.445    | 0.362    | 0.149    | 0.361    | 0.41     | 0.278    | 0.213    | 0.161    | 0.471    |
| DiMP50 [21]      | 0.407    | 0.256    | 0.141    | 0.474    | 0.358    | 0.122    | 0.382    | 0.437    | 0.271    | 0.205    | 0.138    | 0.527    |
| SiamCAR [54]     | 0.377    | 0.209    | 0.119    | 0.424    | 0.313    | 0.102    | 0.386    | 0.448    | 0.241    | 0.163    | 0.109    | 0.52     |
| CLNet [55]       | 0.384    | 0.209    | 0.111    | 0.389    | 0.303    | 0.0905   | 0.358    | 0.391    | 0.229    | 0.141    | 0.105    | 0.527    |
| SiamFC++ [59]    | 0.377    | 0.2     | 0.109    | 0.401    | 0.316    | 0.0828   | 0.333    | 0.371    | 0.237    | 0.138    | 0.0953   | 0.511    |
| SiamMask [58]    | 0.367    | 0.196    | 0.111    | 0.383    | 0.286    | 0.0832   | 0.322    | 0.375    | 0.218    | 0.137    | 0.0922   | 0.505    |
| SiamRPN [28]     | 0.361    | 0.187    | 0.12     | 0.386    | 0.296    | 0.0843   | 0.316    | 0.373    | 0.204    | 0.13     | 0.0875   | 0.495    |
| SiamR-CNN [31]   | 0.405    | 0.242    | 0.193    | 0.339    | 0.322    | 0.176    | 0.334    | 0.28     | 0.27     | 0.228    | 0.169    | 0.471    |
| GlobalTrack [30] | 0.285    | 0.188    | 0.126    | 0.242    | 0.212    | 0.122    | 0.238    | 0.179    | 0.204    | 0.171    | 0.116    | 0.291    |

From the comparison results in Table V, compared with baseline methods, the PR/NPR scores of MKDNet-SD and MKDNet-IoUD are improved by 1.8%/1.9% and 2.0%/1.9%, respectively, which show that score-level distillation and IoU-level distillation can improve the ability of discrimination and localization of tiny object, respectively. Similarly, the PR/NPR scores of MKDNet-FD and MKDNet-FD-GAN are 1.4%/2.0% and 0.7%/1.4% higher than the baseline method, which show that the proposed feature-level distillation and adversarial loss can effectively enhance the representation ability of tiny object features. Compared with MKDNet-FD-GAN, the PR/NPR/SR scores of MKDNet-FD are improved by 0.7%/0.6%/0.7%, which verify that the feature distillation module combining adversarial loss and feature consistency loss is more effective. By observing the performance of MKDNet-noRDM, it can be seen that although the PR/SR scores of MKDNet-noRDM have a certain improvement compared to the baseline method, its performance is much lower than that of MKDNet, indicating that removing RDM will significantly drop the overall performance. In addition, compared with other versions, the MKDNet achieves the best results, which shows that the three distillation strategies proposed jointly can achieve better tracking performance.

To explore the sample imbalance problem, we introduce focal loss [66] to our classification loss. The formula is as...
follows:

\[
I(s, z) = \begin{cases} 
\alpha (1-s) \gamma (s-z), & z > T \\
(1-\alpha) s^{\gamma} \max(0, s), & z \leq T
\end{cases}
\]  

(10)

where \( I(s, z) \) corresponds to \( l(\cdot) \) in (7). \( T \) is a threshold. The region of \( z > T \) is considered as an object, otherwise is background. For the object region \( z > T \), we take the difference between the predicted confidence score \( s \) and the label \( z \), while we only penalize positive confidence values for the background \( z \leq T \). In addition, \( \alpha \in [0, 1] \) is a weighting factor used to balance the importance of positive/negative examples. \( \gamma \geq 0 \) is a tunable focusing parameter. The modulating factor \( (1-\alpha) s^{\gamma} \) is used to solve the problem of unbalanced hard/easy examples. The modulating factor is essentially a hard example mining technique, which gives a small weight to easy examples (that is, examples with high confidence score) and a large weight to hard examples. Following focal loss, we set \( \alpha = 0.25 \) and \( \gamma = 2 \), respectively.

From Table V, we can find that the performance of our tracker has decreased significantly after adding focal loss. The possible reason is that the modulating factor excessively depends on the predicted confidence score, which leads to the overconfidence of the network and affects the generalization of the model.

### E. Impact of Training Data

1) Impact on LaTOT Testing Set: To show the effects of our training set on the performance of trackers, we load the training model of the trackers and retrain MKDNet and other two trackers (Super-DiMP and ATOM). Table VI reports the results of MKDNet, Super-DiMP, and ATOM on LaTOT testing set and comparisons with the performance of original trackers trained on original tracking datasets. It should be noted that we have not changed any hyperparameters of three trackers. From Table VI, we observe that the three trackers gain consistent performance gains, which show the importance of the training set for tiny object tracking.

In addition, it is worth noting that our proposed MKDNet requires high-resolution and low-resolution image pairs to train student network. The LaTOT has no corresponding high-resolution image to guide the learning of low-resolution tiny object images; therefore, we directly load our trained MKDNet model to fine-tune student network on LaTOT training. From the results, we can see: 1) after fine-tuning on LaTOT training set, the performance of our proposed MKDNet has been significantly improved and 2) compared with the baseline method, the PR/NPR/SR scores of MKDNet are improved by 1.7%/0.2%/0.5%. Although the improvement is small, our method far outperforms the baseline method without the tiny object training set.

2) Impact on GOT-10k Testing Set: We retrain the Super-DiMP on LaTOT and the training set of LaSOT and GOT-10K by loading the pretrained model of Super-DiMP. For the sake of fairness, all training and inference parameters are kept the same in the experiments. Table VII shows the tracking results on GOT-10k testing set. Compared with the performance of the original Super-DiMP trained on the training set of LaSOT, GOT-10k, TrackingNet, and MS COCO [49], our results are significantly improved. It proves that our dataset can effectively boost the performance of deep trackers.

### F. Qualitative Evaluation

To intuitively observe the advantages of our tracker compared to other tracking methods, we qualitatively evaluate seven representative trackers, including ATOM, Ocean, Siam R-CNN, ECO, MDNet, SiamMask, and GlobalTrack. In Fig. 7, we show four tracking scenarios with seven challenge attributes, which include low resolution, fast motion,
Fig. 7. Qualitative evaluation on eight representative trackers. To facilitate observation, we enlarge object regions and display it on the right of original images. (a) Badminton4. (b) Bird17. (C) Poker_box.

Fig. 8. Failed cases of our tracker and other state-of-the-art trackers. (a) Golf2. (b) Pingpang1. (c) Badminton9. (d) Car23.

out-of-view, motion blur, similar object, full occlusion, and abrupt motion. Moreover, to facilitate observation, we enlarge the target object regions and display them on the right of original image. On LaTOT, multiple challenge attributes often appear in the same sequences, which bring a huge challenge for tracking tiny objects and easily lead to failures of current trackers. For example, as shown in Fig. 7(a), the sequence of bidminton4 has the challenges of fast motion, motion blur, out-of-view, and low resolution, making it difficult for current trackers. Thanks to our algorithm, it can significantly improve the feature representation, discrimination, and localization abilities for the tiny object to solve these complex challenges compared with other advanced trackers.

G. Failed Cases

As shown in Fig. 8, we show some failed cases of our tracker and other state-of-the-art trackers on LaTOT. In these failed cases, there are six challenge attributes, including abrupt motion [as shown in Fig. 8(a)], motion blur [as shown in Fig. 8(b) and (c)], out-of-view [as shown in Fig. 8(c)], full occlusion [as shown in Fig. 8(d)], fast motion [as shown in Fig. 8(a), (b), and (c)], and background clutter [as shown in Fig. 8(c) and (d)]. The challenging problems of abrupt motion, fast motion, and out-of-view will cause target to exceed the search area of the trackers. Even if there are some redetection tracking algorithms that can solve this kind of problem, because the tiny object lacks enough appearance information and the background interference is too large, these trackers cannot track the object again. Motion blur is also a big challenging problem in LaTOT, which is often accompanied by the challenging attributes of fast motion and camera motion. It will directly affect the representation quality of features. As show in Fig. 8(d), the challenge attributes of full occlusion can easily lead to model drift and targets beyond the search area. To sum up, the main reasons why our tracker and other trackers fail to track are summarized as follows: 1) tiny object have low resolution, more noise, and less effective information, which make trackers unable to effectively extract its features and accurately locate the target and 2) on LaTOT, there are often multiple challenge attributes in the same video sequence, which brings a huge challenge to tracking methods.

VI. CONCLUSION AND FUTURE WORKS

In this article, we propose an MKDNet to effectively enhance the feature representation, discrimination, and localization abilities for tiny objects in visual tracking. This method
includes three levels of feature-level distillation, score-level distillation, and IoU-level distillation to significantly boost the performance of tiny object tracking. To avoid wrong and invalid distillation, a reliable distillation measure is also introduced to control the distillation process. To provide a comprehensive evaluation platform, we present a unified benchmark with high-quality dense annotations, high diversity, and challenges for LaTOT. Moreover, we set 12 challenge attributes to evaluate and analyze trackers. Extensive experiments are performed on the proposed dataset, and the results prove the superiority and effectiveness of MKDNet compared with state-of-the-art tracking methods. The results also show that current trackers still have a large research space in tiny object tracking.

In the future, some potential directions for tiny object tracking can be considered. We find from Table IV that the challenges of fast motion, partial occlusion, full occlusion, and motion blur are greatly damaging to the performance of tiny object tracking, which shows that it is difficult to estimate positions of tiny objects directly through limited appearance information. Therefore, we can employ some trajectory prediction algorithms [67] to estimate locations of tiny objects by combining the historical location information and appearance information. Taking the fast motion challenge as an example, trajectory prediction can help trackers determine the range of target searching area, so as to help accurately locating the target. In addition, optical flow methods [68] can also be explored to predict target appearance states and locations in future frames, thereby help solving the problems of motion blur and fast motion.

REFERENCES

[1] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, “Action-driven visual object tracking with deep reinforcement learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 6, pp. 2239–2252, Jun. 2018.
[2] F. Zhao, T. Zhang, Y. Wu, M. Tang, and J. Wang, “Antidelay LSTM for Siamese tracking with adversarial learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 10, pp. 4475–4489, Oct. 2021.
[3] S. Ge, C. Zhang, S. Li, D. Zeng, and D. Tao, “Cascade correlation refinement for robust deep tracking,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 10, pp. 4475–4489, Oct. 2021.
[4] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, “Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9725–9734.
[5] M. Kisantall, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” 2019, arXiv:1902.07296.
[6] D. Guo, L. Zhu, Y. Lu, H. Yu, and S. Wang, “Small object sensitive segmentation of urban street scene with spatial adjacency between object classes,” IEEE Trans. Image Process., vol. 28, no. 6, pp. 2643–2653, Jun. 2019.
[7] H. Wang, L. Zhou, and L. Wang, “Miss detection vs. False alarm: Adversarial learning for small object segmentation in infrared images,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 8509–8518.
[8] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 2411–2418.
[9] Y. Wu, J. Lim, and M. H. Yang, “Object tracking benchmark,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1834–1848, Sep. 2015.
[10] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, and Z. Chi, “The visual object tracking VOT2016 challenge results,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 1–16.
[11] M. Kristan et al., “The sixth visual object tracking VOT2018 challenge results,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 1–22.
[12] M. Kristan et al., “The visual object tracking VOT2015 challenge results,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2015, pp. 1–23.
[13] M. Kristan et al., “A novel performance evaluation methodology for single-target trackers,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 11, pp. 2137–2155, Nov. 2016.
[14] M. Kristan et al., “The seventh visual object tracking VOT2019 challenge results,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops, Oct. 2019, pp. 1–36.
[15] S. Hadfield, K. Lebeda, and R. Bowden, “The visual object tracking VOT2014 challenge results,” in Proc. Eur. Conf. Comput. Vis. ECCV) Vo. Object Tracking Challenge Workshop, 2014, pp. 777–823.
[16] M. Müller, A. Bibi, S. Giancola, S. Alsulbahi, and B. Gahmen, “TrackingNet: A large-scale dataset and benchmark for object tracking in the wild,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 300–317.
[17] H. Fan et al., “LaSOT: A high-quality benchmark for large-scale single object tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 5374–5383.
[18] L. Huang, X. Zhao, and K. Huang, “GOT-10 k: A large high-diversity benchmark for generic object tracking in the wild,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1562–1577, May 2021.
[19] C. Liu et al., “Aggregation signature for small object tracking,” IEEE Trans. Image Process., vol. 29, pp. 1127–1129, Oct. 2020.
[20] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4293–4302.
[21] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, “Learning discriminative model prediction for tracking,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6182–6191.
[22] L. Wang and K.-J. Yoon, “Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 6, pp. 3048–3068, Jun. 2022.
[23] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanee-Yakhdan, and S. Kasaei, “Deep learning for visual tracking: A comprehensive survey,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 5, pp. 3943–3968, May 2022.
[24] K. Tong, Y. Wu, and F. Zhou, “Recent advances in small object detection based on deep learning: A review,” Image Vis. Comput., vol. 97, May 2020, Art. no. 103910.
[25] J. Jung, J. Son, M. Baek, and B. Han, “Real-time MDNet,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 83–98.
[26] X. Wang, J. Tang, B. Luo, Y. Wang, Y. Tian, and F. Wu, “Tracking by joint local and global search: A target-aware attention-based approach,” IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 11, pp. 6951–6945, Nov. 2022.
[27] X. Wang et al., “Dynamic attention guided multi-trajectory analysis for single object tracking,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 12, pp. 4895–4908, Dec. 2021.
[28] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with Siamese region proposal network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8971–8980.
[29] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “SiamRPN++: Evolution of Siamese visual tracking with very deep networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4282–4291.
[30] L. Huang, X. Zhao, and K. Huang, “GlobalTrack: A simple and strong baseline for long-term tracking,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 7, pp. 11037–11044.
[31] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, “Siam R-CNN: Visual tracking by re-detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6578–6588.
[32] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “ATOM: Accurate tracking by overlap maximization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4660–4669.
[33] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, “Acquisition of localization confidence for accurate object detection,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 784–799.
[34] S. M. Marvasti-Zadeh, J. Khaghanii, H. Ghanee-Yakhdan, S. Kasaei, and L. Cheng, “COMET: Context-aware IoU-guided network for small object tracking,” 2020, arXiv:2006.02597.
[35] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1501.02025.
Xiao Wang (Member, IEEE) received the B.S. degree from West Anhui University, Lu’an, China, in 2013, and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 2019. From 2015 to 2016, he was a Visiting Student with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He has visited the UBTECH Sydney Artificial Intelligence Centre, Faculty of Engineering, The University of Sydney, Sydney, NSW, Australia, in 2019. He was a Post-Doctoral Researcher with the Peng Cheng Laboratory, Shenzhen, China, in 2019 and 2021. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. His current research interests include computer vision, machine learning, pattern recognition, and deep learning.

Jin Tang received the B.Eng. degree in automation and the Ph.D. degree in computer science from Anhui University, Hefei, China, in 1999 and 2007, respectively. He is currently a Professor with the School of Computer Science and Technology, Anhui University. His current research interests include computer vision, pattern recognition, and machine learning.

Bin Luo (Senior Member, IEEE) received the B.Eng. degree in electronics and the M.Eng. degree in computer science from Anhui University, Hefei, China, in 1984 and 1991, respectively, and the Ph.D. degree in computer science from the University of York, York, U.K., in 2002. He is currently a Professor with Anhui University. He has authored over 200 papers in journals, edited books, and refereed conferences. His current research interests include random graph-based pattern recognition, image and graph matching, graph spectral analysis, and video analysis. Dr. Luo also chairs the IEEE Hefei Subsection. He served as a Peer Reviewer of international academic journals, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Pattern Recognition, Pattern Recognition Letters, the International Journal of Pattern Recognition and Artificial Intelligence, Knowledge and Information Systems, and Neurocomputing.

Zhixiang Huang (Senior Member, IEEE) was born in Anhui, China, in 1979. He received the B.S. and Ph.D. degrees from Anhui University (AHU), Hefei, China, in 2002 and 2007, respectively. Since 2008, he has been a Full Professor with the School of Electronic Information and Engineering, AHU. From September 2010 to September 2011, he was a Visiting Scholar with Iowa State University, Ames, IA, USA. From August 2013 to October 2013, he was a Visiting Professor with The University of Hong Kong, Hong Kong. From February 2014 to February 2015, he was a Visiting Professor with the Beijing National Laboratory for Condensed Matter Physics, Institute of Physics, Chinese Academy of Sciences, Beijing, China. He has coauthored one monograph on the symplectic finite-difference time-domain method and two book chapters for CRC Press and InTech publishers. He has coauthored 90 peer-reviewed journal articles included in the Web of Science Core Collection. His current research interests include time-domain numerical methods, metamaterials, and active metamaterials. Dr. Huang was a recipient of the Second Prize of Science and Technology from the Anhui Province Government, China, in 2015, and the National Science Foundation for Outstanding Young Scholar of China in 2017.