Localization-Aware Meta Tracker Guided with Adversarial Features

YITING LIN¹, BINENG ZHONG¹, GUORONG LI², SICHENG ZHAO³, ZIYI CHEN¹ AND WENTAO FAN¹

¹Department of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, China (e-mail: 17013083006@hqu.edu.cn; bnzhong@hqu.edu.cn; chenziyi@hqu.edu.cn; fwt@hqu.edu.cn)
²School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China (email: liguorong@ucas.ac.cn)
³Berkeley Artificial Intelligence Research Lab, University of California Berkeley, USA (email: schzhao@gmail.com)

Corresponding author: Bineng Zhong (e-mail: bnzhong@hqu.edu.cn).

This work is supported by Natural Science Foundation of China (No. 61572205, 61802135, 61876068, and 61701273), Natural Science Foundation of Fujian Province (No. 2017J01113, and 2019J01081).

ABSTRACT

Deep learning has recently shown great potentials in learning powerful features for visual tracking. However, most of the deep learning based trackers neglect localization accuracy in the evaluation process of candidates. What’s more, they usually over-rely on the discriminative features in a single frame in the training process. Consequently, they may fail when the discriminative features are occluded or changed in the tracking phase. In this work, we propose a novel localization-aware meta tracker guided with adversarial features (named LMT) to address the above issues. First of all, we design a novel Intersection over Union guided method to effectively balance the problem of classification and localization accuracy. To further improve the robustness of our classifier, we creatively use adversarial features during offline training phase. Those adversarial features can effectively guide the classifier in learning how to better deal with the situation where the discriminative features are occluded or changed. Finally, benefiting from meta learning, our algorithm only needs to perform one iterative update on the first frame and it can perform well on the tracking scenarios. Extensive experiments demonstrate the outstanding performance of our LMT tracker compared with state-of-the-art trackers on three benchmarks: OTB-2015, VOT-2016, and VOT-2018.

INDEX TERMS adversarial features, localization accuracy, visual object tracking

I. INTRODUCTION

VISUAL object tracking is an important research field in computer vision, which has been widely applied in video surveillance, driverless driving, robotics and so on. Traditional methods have achieved promising performances from different aspects, such as subspace learning [34], particle filter framework [36], compressive learning [37], weakly supervised learning [48], contour [35] based and biologically inspired [44] appearance models. However, the performance is still limited by hand-engineered features or low-level features. In recent years, with the advancement of deep learning, deep learning based trackers have further improved tracking performance [13], [41], [43], [46], [47] on multiple standard benchmarks [1]–[4]. Despite the demonstrated success, there are still two issues that have not been well resolved.

Firstly, prior trackers [5], [6] have made limited efforts on improving localization accuracy. These trackers only

**FIGURE 1.** Some representative candidate patches with classification confidence scores and their corresponding IoU from (a) MetaSDNet and (b) our LMT tracker, respectively. In our LMT tracker, the candidate with the highest classification confidence also has the highest IoU. In other words, our LMT tracker can achieve a better balance between the classification task and localization task.
use classification confidences to evaluate candidate patches. However, a candidate patch with the highest classification confidence may not be the most accurate one. As shown in Fig. 1 (a), the classification confidence scores from MetaSD-Net [5] are inconsistent with their corresponding Intersection over Union (IoU). Therefore, we use an IoU-guided method to improve this problem. We add the prediction information of localization accuracy to the evaluation process of the candidate patches. In Fig. 1 (b), our LMT tracker achieves a better balance between classification and localization tasks.

Secondly, in the training process, these trackers usually over-rely on the discriminative features in a single frame. Consequently, they may fail when the discriminative features are occluded or changed in the tracking phase. There have been some attempts [9]–[11] to enhance the robustness of the tracking algorithms using adversarial learning. These trackers have tried to generate images that are consistent with the distribution of tracking sequences or generate rich appearance changes. However, they may introduce noise that is inconsistent with the sequence distribution during the model updating process. Different from above trackers, we generate adversarial features to improve the robustness of our classifier in offline training phase. As shown in Fig. 1, the generated adversarial features suppress strong discriminative part and enhance the weak discriminative part in the original features. Therefore, the classifier will pay more attention to other useful features in the target object. What’s more, we use meta learning as our training method so that the model can quickly fit the current object.

The main contributions of this work are as follow:

1) We propose an IoU-guided method to reweight the classification confidence scores of the candidate patches. Therefore, we can strike a balance between classification and localization tasks, and thus improve the localization and classification performance simultaneously.

2) We creatively employ adversarial features to guide the training process of our classifier, which will pay attention to more useful features except the most discriminating feature in current frame. Therefore, our LMT tracker can better deal with various appearance variations.

3) We use meta offline learning to train the localization and classification tasks. Consequently, we can quickly fit the current object quickly using only one iterative update on the first frame.

II. RELATED WORK

Visual tracking has been an active research topic over the last decades. A comprehensive survey of the related trackers is beyond the scope of this paper. State-of-the-art trackers are mainly based on the correlation filter frameworks [40] and the tracking-by-detection frameworks [37], [39], [41], [42], [45]. Please refer to [7], [8] for more complete reviews on tracking literature. In this section, we provide a brief overview of related trackers, i.e., improving localization accuracy based trackers, adversarial learning based trackers, and meta learning based trackers.

A. IMPROVING LOCALIZATION ACCURACY BASED TRACKERS

In terms of improving localization accuracy, most approaches [12], [40] naively adopt a multi-scale strategy with its obvious computational impact and sensitive to scale. SiamRPN [13] has shown the ability of bounding box regression to further improve the localization accuracy. ATOM [14] proposes a novel multi-task tracking framework with explicit components for target estimation and classification. However, ATOM use classifier trained online directly, which may lead to relatively poor robustness. Unlike above mentioned works, we combine the IoU-guided method with adversarial learning to improve both localization accuracy and robustness of our LMT tracker.

B. ADVERSARIAL LEARNING BASED TRACKERS

Inspired by the success of adversarial learning [15], there have been some attempts to enhance the robustness of tracking algorithms using adversarial learning [9]–[11]. SDCFAL [11] and VITAL [10] use generative adversarial networks to enhance the positive samples in the feature space, enrich the object appearance changes, and thus update the tracker model more robustly. But the noise generated by online learning may lead to erroneous updates. SINT++ [9] generates images with less frequent occurrence in tracking sequences to narrow the gap between the training set and the test set. The generated data is expected to appear in the test set. Some
adversarial attacks based methods [21], [38] attack deep networks on image classification, semantic segmentation, and object detection by generating adversarial samples with visually imperceptible perturbations. However, there is still no guarantee that the samples generated by SINT++ or adversarial attacks based methods are inconsistent with the tracking sequences. Unlike above mentioned works, we use a fast gradient method to generate adversarial samples in the feature space. The adversarial features suppress the strong discriminative part and enhance the weak discriminative part in the original features, which are in line with our need to train robust classifiers.

C. META LEARNING BASED TRACKERS
Meta learning, also called learning to learn, aims to effectively fine-tune the network based on few given training examples. Inspired by the success of meta learning [16]–[18], there have been some attempts to use meta learning in the field of tracking because of its fast learning nature. MLT [19] adds the instance-level information constructed by the meta-learning method to the Siamese network, which aims to customize the feature space for a specific target and accurately track the target. Meta-Tracker [5] uses the meta learning method to obtain initialization parameters, learning rate, and update direction during offline training, which can guide the model to converge faster and better. However, these trackers still cannot deal well with various challenges in tracking phase due to the insufficient training data. Therefore, we combine the method of adversarial learning with meta learning to alleviate poor robustness problem. The adversarial features generated by adversarial learning can effectively compensate for the lack of training data.

III. THE PROPOSED TRACKER
In this work, our proposed tracker consists of feature extraction network, classification network, and IoU prediction network. The framework of our LMT tracker is illustrated in Fig. 3. More specifically, our LMT tracker works as follows: given the first frame of a new sequence, we perform one iterative updating to quickly adapt the pre-trained models to an interested target object. Then we train a simple linear regression model to predict the precise target location using the features of the samples near the target location. In the subsequent frames, some candidate patches collected from the current frame are firstly cropped. And we use the classification network to calculate confidence scores of the candidate patches. Then, based on our IoU-guided method, we will get the adjusted classification confidences. Next, we will select top K candidate patches according to the adjusted classification confidences and further get the localization of the object. Finally, we use the regression model to fine-tune the target locations if the estimated targets are reliable [20].

In the offline training phase, we use multi-task meta training method and adversarial meta training method to train the three networks in our framework. In the following, we will describe the details for multi-task meta learning, adversarial learning, and IoU-guided tracking, respectively.

A. MULTI-TASK META TRAINING FOR OUR TRACKER
In this subsection, we describe how to train all of the three networks in our framework by multi-task meta training. The whole algorithm is illustrated in Algorithm 1. We define feature extraction network, classification network, and IoU prediction network as $F(\cdot)$, $C(\cdot)$, and $R(\cdot)$, respectively. We use Adam as the optimizer in the offline training phase. The offline meta training of our algorithm can be described as two stages. In stage 1, we use current frame to iteratively update the initial weight $\theta_0$ to get $\theta_1$. The update function $M$ is parameterized by $\alpha$:

$$M(\theta, \nabla_\theta L) = \theta - \alpha \odot \nabla_\theta L,$$

where $\theta$, $\alpha$, $L$ and $\odot$ denotes the initial weight, initial learning rate, loss function and element-wise product, respectively. The size of $\alpha$ is the same as the tracker parameters $\theta$. The sampling process is shown in line 4 of Algorithm 1. $j$, $j_1$ and $\delta$ denote the current frame, the future frame and the interval between the future frame and the current frame, respectively. As described in line 12 of Algorithm 1, we
Algorithm 1 Multi-task Meta Training Algorithm for Our Tracker

Input: Randomly initialize $\theta_0$ and $\alpha$, training dataset $D$

Output: $\theta_0^*$ and $\alpha^*$

1: while not converge do
2: $\text{grad}_{\theta_0}, \text{grad}_{\alpha} = 0$ $\triangleright$ Initialize to zero vector
3: for all $k \in \{0,...,N_{\min} - 1\}$ do
4: $x, j, \delta \sim P(D)$ $\triangleright$ Sample a training example
5: $\theta_0^k = \theta_0$
6: Stage 1: update the initial weight
7: for all $t \in \{0,...,T-1\}$ do
8: $\text{Feat}_{j1} = F(x_j, \theta_0^t)$
9: $\hat{C}_{j1} = C(\text{Feat}_{j1}, \theta_0^t)$
10: $\hat{R}_{j1} = R(\text{Feat}_{j1}, \theta_0^t)$
11: $\theta_0^{t+1} = \theta_0^t - \alpha \odot \nabla_{\theta_0} L_{\text{all}}(C_{j1}, \hat{C}_{j1}, \hat{R}_{j1}, \hat{R}_{j1}; \theta_0^t)$
end for
12: $\theta_1 = \theta_0^T$
13: Stage 2: optimize $\theta_1$ and $\alpha$
14: $\text{Feat}_{j1} = F(x_j, \theta_1)$ $\triangleright$ $j_1 = j + \delta$
15: $\hat{C}_{j1} = C(\text{Feat}_{j1}, \theta_1)$
16: $\hat{R}_{j1} = R(\text{Feat}_{j1}, \theta_1)$
17: $\text{grad}_{\theta_0} = \text{grad}_{\theta_0} + \nabla_{\theta_0} L_{\text{all}}(C_{j1}, \hat{C}_{j1}, \hat{R}_{j1}, \hat{R}_{j1})$
18: $\text{grad}_{\alpha} = \text{grad}_{\alpha} + \nabla_{\alpha} L_{\text{all}}(C_{j1}, \hat{C}_{j1}, \hat{R}_{j1}, \hat{R}_{j1})$
end for
19: $\theta_0 = \text{Optimizer}(\theta_0, \text{grad}_{\theta_0})$ $\triangleright$ Update $\theta_0$
20: $\alpha = \text{Optimizer}(\alpha, \text{grad}_{\alpha})$ $\triangleright$ Update $\alpha$
21: end while

use current frame to iteratively update the initial weight $\theta_0$ once to get $\theta_1$. Therefore, the model can pay more attention to current object. By enforcing the number of update iterations during meta-training, the training of the model becomes significantly faster during the initialization. Meanwhile, it can effectively study how to adapt to new target with one update. In stage 2, $\theta_1$ is used to predict the state in the future frame. Benefiting from the two-stage training, our algorithm can cope with future changes better. In previous trackers [6], [10], the learning rate is manually set. In our algorithm, we optimize the initial parameters $\theta$ and the learning rate $\alpha$ simultaneously. It is obvious that the $\odot$, convolution, pool and full connection operation are all differentiable. Therefore, the update function is differentiable. $\alpha$ can guide each parameter in updating with appropriate learning rate and direction. And the model can adapt to the new target quickly in the tracking phase.

In multi-task meta training, the loss function $L_{\text{all}}$ consists of classification loss $L_{\text{cla}}$ and regression loss $L_{10U}$. Classification task tends to be easier to train than IoU prediction task, which may lead to insufficient learning in the IoU prediction section. Therefore, we propose an attentive loss function to balance the loss of both tasks, making their values consistent.

Our loss function is shown as follow:

$$L_{\text{all}} = L_{\text{cla}} + \lambda \ast L_{10U},$$

(2)

$$L_{10U} = L_{\text{Pos10U}} + L_{\text{Neg10U}},$$

(3)

$$L_{\text{Pos10U}} = \text{MSE}(\beta \ast (\text{PR}_{10U}, \text{PR}_{10U})), $$

(4)

$$L_{\text{Neg10U}} = \text{MSE}(\beta \ast (\text{NR}_{10U}, \text{NR}_{10U})), $$

(5)

where $\lambda$ is a balance factor used to balance the relationship between the classification and the localization tasks. $\beta$ is an expansion factor, which is multiplied to the predicted IoU value and its corresponding label to prevent the insufficient training of the IoU prediction network. The $\text{PR}_{10U}$ and $\text{NR}_{10U}$ are the IoU predictive value of positive and negative samples, respectively. $\text{PR}_{10U}$ and $\text{NR}_{10U}$ are corresponding groundtruth, respectively. The labels are obtained by calculating the overlap between the candidate patches and the groundtruth. $\text{PR}_{10U}$ and $\text{NR}_{10U}$ are collectively called $R$. $\text{PR}_{10U}$ and $\text{NR}_{10U}$ are collectively called $C$. $C$ and $R$ denote the predicted classification confidence scores and corresponding groundtruth, respectively. $\text{MSE} (\cdot)$ means square error loss.

B. ADVERSARIAL META TRAINING FOR OUR TRACKER

In this subsection, we describe how to further fine-tune the parameters of the classification network by adversarial meta training. After multi-task meta training, the parameters of feature extraction and IoU prediction network are frozen. Only the parameters of classification network will be fine-tuned in adversarial meta learning phase. We use a fast gradient method to generate adversarial features, and then use them in training phase.

FGSM [21], also called fast gradient sign method, is a classic method to generate adversarial samples. This method generates adversarial samples as follows:

$$x_{adv} = x + \eta,$$

(6)

where $\eta$ can be calculated as follows:

$$\eta = \epsilon \ast \text{sign}(\nabla_x J(\theta, x, y)),$$

(7)

where $J$ and $x$ denote the loss function and input sample, respectively. $\text{sign}(\nabla J)$ describes the gradient direction of the sample $x$, and $\epsilon$ is the magnitude of the offset in that direction. Specially, the gradient direction at $x$ will always point to the direction in which the loss function increases. Hence, FGSM is the easiest and most effective way to generate adversarial samples. However, directly applying FGSM in the tracking problem is not feasible. Firstly, FGSM produces samples with little perturbations imperceptible to human, which are inconsistent with the tracking sequences. Secondly, only the direction of the gradient is considered and the gradient amplitude is not fully utilized to adjust the input samples. Therefore, we propose an improved gradient-based method to generate adversarial features. Instead of adding imperceptible perturbations directly, we use the gradient information to suppress or enhance the features. Therefore, the classifier will
be forced to seek more useful features in the meta offline training. Algorithm 2 shows the overall process of adversarial training. Stage 1 is the same as multi-task meta training phase.

As described in line # 18 of Algorithm 2, Adv(·) is adversarial feature generation module, which serves as a weight to decide which type of features should be dominant. Adv(·) contains a tanh activation function, calculation of the weight of the features and uses the weights to enhance or suppress different parts of features. Adversarial features can be generated as follows:

\[
\text{grads}_{\text{Feat}_j} = \text{tanh}(\text{grads}_{\text{Feat}_j}), \quad (8)
\]

\[
\text{weight}_{j} = I + \epsilon \times \text{grads}_{\text{Feat}_j}, \quad (9)
\]

\[
\text{AdvFeat}_j = \text{Feat}_j \times \text{weight}_{j}, \quad (10)
\]

where \(\epsilon\) denotes the step size that controls the influence of gradient information. The tanh layer normalizes the \(\text{grads}_{\text{Feat}_j}\) to \([-1, 1]\). \(I\) is a matrix with pixel values of all ones and it is the same size as the \(\text{grads}_{\text{Feat}_j}\). The gradient direction will always point to the direction in which the loss function increases, and the absolute value of the gradient indicates the rate of change. If \(\text{grads}_{\text{Feat}_j}\) is negative, the value of weight will be smaller than 1, so the weight of this part is suppressed. If \(\text{grads}_{\text{Feat}_j}\) is positive, the value of weight will be larger than 1, so this part is enhanced. Hence, the generated adversarial features will maximize the classification difficulty of the classifier.

Our adversarial meta learning method has two advantages. First, the method to generate adversarial features is simple and effective. It only performs one step gradient update along the direction of gradient at each pixel. Second, the generated adversarial samples can suppress the part with strong discriminative power and enhance the weak discriminative part. Therefore, the candidate patches can be correctly classified even when the discriminative features are suppressed through this adversarial learning method. Finally, combined with meta-learning training method, the model can quickly adapt to current object in the tracking phase.

### C. IOU-GUIDED TRACKING

Our proposed tracker works as follows in tracking phase:

**Step 1**: The input of tracking algorithm is candidate patches collected from the current frame. The features are extracted through the feature extraction network. Then we use these features to obtain IoU prediction values \(\hat{R}\) and the classification confidence scores \(\hat{S}_{\text{ori}}\) through IoU prediction network and classification network, respectively.

**Step 2**: If the classification confidence is smaller than 0, no adjustment is made. This is because candidate patches with negative confidence scores have little effect on the final result. If the samples have positive confidence scores, we use the \(\hat{R}\) to adjust them and obtain \(S_{\text{adj}}\) as follows:

\[
S_{\text{adj}} = S_{\text{ori}} \ast \hat{R}. \quad (11)
\]

For the candidate patches with high classification confidence scores and low IoU prediction values, the confidence scores will be reduced. Therefore, \(S_{\text{adj}}\) can better reflect the localization performance of candidate patches.

In addition, if \(\hat{R} < 0.3\): \(S_{\text{adj}} = \hat{Q}\).

The classification confidences of candidate patches with low IoU prediction values will be set to \(\hat{Q}\), a scalar with small value. Therefore, the inaccurate candidate patches can be further excluded.

**Step3**: The motion of two adjacent frames in target tracking sequence is relatively smooth, so we further exclude samples with low overlap of two consecutive frames. Starting from the second frame, we calculate the IoU value between each candidate sample and the prediction result of the previous frame: \(C_{\text{IoU}}\), if \(C_{\text{IoU}} < 0.1\): \(S_{\text{adj}} = \hat{Q}\).

**Step4**: After the above three steps, we obtain the adjusted classification confidences \(S_{\text{adj}}\). Then we select the top \(K\) according to \(S_{\text{adj}}\) and calculate their mean location.

**Step5**: Finally, we apply the bounding box regression technique to further improve target localization accuracy if the estimated targets are reliable, i.e., \(S_{\text{adj}} > 0\). And we obtain the final target location after bounding box regression.
In the OTB-2015, we use the one-pass evaluation (OPE) with precision and success plots metrics. In the VOT-2016 and VOT-2018, we measure the performance in terms of Expected Average Overlap (EAO) and Accuracy-Robustness (A-R) ranks.

### C. EXPERIMENTAL RESULTS

1) **OTB-2015 Dataset**

We compare our LMT tracker on the OTB-2015 benchmark with state-of-the-art trackers, including VITAL [10], MDNet [6], MetaSDNet [5], ECO [25], CCOT [26], CNN-SVM [27], SRDCFdecon [28], Staple [29] and MCPF [32]. Fig. 4 shows the results from all compared trackers. Overall, our tracker performs favorably against state-of-the-art trackers in both localization precision and overlap success. Fig. 5 compares the performance over eight video attributes using one-pass evaluation. Our tracker handles large appearance variations caused by deformation, in-plane and out-of-plane rotations better. We attribute our performance improvement to the adversarial features guided training and IoU-guided tracking, which can alleviate the inaccurate localization problem. Adversarial training uses features with weak discriminative to train the classifier. Therefore, we can also effectively classify through other useful features when appearance changes happen. In our method, the weak discriminative part of adversarial features can provide effective information for the classifier in most cases, which can improve the robustness of the classifier. However, these weak discriminative parts will be invalid in a few cases. For example, when full occlusion occurs, we cannot obtain any valid information from the extracted features of the candidate patches. Actually, full occlusion is always a big challenge in tracking literature. We will further address this issue in the future by incorporating a re-detection model into our LMT tracker.

2) **VOT Dataset**

We compare our LMT tracker with state-of-the-art trackers on the VOT-2016 and VOT-2018 benchmarks. Table 1 shows the EAO value of these trackers. Fig. 6 shows the A-R plots on VOT-2016 and VOT-2018. In VOT-2016, the A-R performance of our LMT tracker is comparable to ECO and better than MDNet and MetaSDNet. In VOT-2018, our performance is significantly better than several other trackers.

### B. EVALUATION METRICS

TABLE 1. The EAO results on VOT-2016 and VOT-2018.

<table>
<thead>
<tr>
<th></th>
<th>ECO</th>
<th>LMT</th>
<th>CCOT</th>
<th>VITAL</th>
<th>MetaSDNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>VOT-2016</td>
<td>0.374</td>
<td>0.337</td>
<td>0.331</td>
<td>0.323</td>
<td>0.314</td>
</tr>
<tr>
<td>VOT-2018</td>
<td>0.280</td>
<td>0.309</td>
<td>0.267</td>
<td>0.144</td>
<td>0.079</td>
</tr>
</tbody>
</table>

### IV. EXPERIMENTS

In this section, we firstly present the experimental settings. Then, we compare our LMT tracker with state-of-the-art trackers on the benchmark datasets OTB-2015 [2], VOT-2016 [3] and VOT-2018 [4]. Moreover, we perform an ablation study on several components of our LMT tracker. Finally, we evaluate how the parameter variations affect our LMT tracker.

#### A. THE EXPERIMENTAL SETTING

During the offline training phase, we use the meta training method to train the parameters in our model. We use Adam as our optimizer in both the multi-task training and adversarial training phases. The parameters of $Q$, $K$, and $S$ are set to -5, 5, and 256, respectively.

1) **Dataset for Meta Offline Training**

We use a large scale video detection dataset [23] and a high-quality benchmark LaSOT [24] for meta offline training. Be similar to [5], we choose 718 video sequences in the ImageNet dataset. Sequences in LaSOT are selected with the probability 0.3. We sample a randomly starting frame from a random video as the current frame, and a random frame after the current frame in the same sequence as the future frame. The interval between the future frame and the current frame is $\delta$. The larger $\delta$, the larger target object variations and environment changes are incorporated into training process.

2) **Meta Offline Training Details**

Our meta offline training can be divided into two phases: multi-task training phase and adversarial training phase. We use the first three conv layers from pre-trained VGG-M [22] as our feature extraction network. During multi-task training phase, we only update the parameters of classification network and IoU prediction network for the first 7K iterations and train all layers for the rest of 13K iterations. We first initialize the learning rate $\alpha$ and learning rate of Adam to 1e-4 and 1e-5, then we reduce them to 1e-5 and 1e-6 after 10K iterations respectively. During adversarial meta training phase, we only train the parameters of classification network. The number of iterations is set to 3K. The parameters of $\alpha$, meta update iteration $T$, step size $\epsilon$, balance factor $\lambda$, mini-batch size $N_{\text{min}}$, expansion factors $\beta$, and the interval between the future frame and the current frame are set to 1e-5, 1, 1, 3, 8, 10, and 10 respectively.
Localization-Aware Meta Tracker Guided with Adversarial Features

3) Qualitative Evaluation

Fig. 7 qualitatively compares the results of 10 trackers. In most sequences, the performance of CNN-SVM is not satisfied because it does not explicitly consider the robustness of its model. MDNet improves CNN-SVM through an end-to-end training method. It performs well on in-plane (Blurowl), and motion blur (Liquor, Blurcar3). MetaSDNet improves MDNet by using meta learning based method to quickly fit to the target object in the initiation phase. It performs well on deformation (Skiing, Bolt2), and motion blur (Deer, Girl2). We can observe that the localization accuracy of our LMT tracker is better than MetaSDNet obviously. For example, the sequence of Basketball contains interference caused by similar objects and dramatic appearance changes. Consequently, some trackers drift to the similar objects or unable to accurately locate. In contrast, our LMT tracker can accurately locate the target object due to the proposed IoU-guided mechanism which is applied to the estimation of candidate patches. Moreover, our adversarial features guided training can make our LMT tracker handles with various changes better.

4) Ablation Studies

There are two important components in our LMT tracker, i.e., multi-task meta training, and adversarial training. To validate the effectiveness of each component, we first implement a MetaTracker-V with our experimental setting to exclude the influence of other factors. The main difference between MetaTracker-V and MetaTracker is that we replace VOT dataset with LaSOT dataset during the training phase, which will improve the generalization ability of our LMT tracker. Then we implement three approaches based on MetaTracker-V to show the effect of each component. We also compare our method with MDNet [6], which has similar network structure compared with our LMT tracker. Analysis and experimental validation with these baseline frameworks will help to differentiate performance improvement. Table 2 shows the ablation studies on OTB-2015 dataset. We observe that these two component have achieved the improvement of 1.1%, and 1.4% in the AUC compared with MetaTracker-V, respectively. And compared with MDNet, they have achieved the improvement of 2.7%, and 3% in the AUC, respectively. The compared results clearly verify that both components make contributions to our LMT tracker. Although a similar network structure is adopted compared with our method, MDNet and MetaTracker only use the classification confidences as the evaluation criterion of the candidate patches, which limits the performance.
FIGURE 7. Qualitative result of ten trackers on twenty challenging sequences from OTB-2015, i.e., Basketball, Bird1, Freeman4, Box, Soccer, Girl2, Bolt2, Blurowl, Dragonbaby, Tiger2, Skiing, Motorrolling, Jogging, Liquor, Blurbody, Blurcar3, Deer, Doll, Faceocc1, and Cardark.
improvement of the localization accuracy. In our multi-task meta training, the IoU prediction network is used to adjust the classification confidences of the candidate patches, which will be helpful to select patches with higher localization accuracy. What’s more, the adversarial meta training further improves the robustness of the classifier.

D. SENSITIVITY TO PARAMETERS
Due to the balance factor $\lambda$ and the expansion factor $\beta$ are crucial parameters in our LMT tracker, one may wonder how they affect our LMT tracker. To answer the question, we check the precision for different parameter settings. Table 3 shows the results of sensitivity analysis on OTB-2015. It is obvious that, for all parameters, a good value can be chosen across a wide range. Moreover, the experiments have shown that a good set of parameters for a video sequence usually performs well for other video sequences (see Fig. 4, Fig. 5, Fig. 6, and Fig. 7).

V. CONCLUSION
In this paper, we have proposed a novel location-aware meta tracker guided with adversarial features. To explicitly address the localization accuracy, we design an IoU-guided method to effectively balance the problem of classification and localization accuracy. Meanwhile, an adversarial training strategy is used to improve the robustness of our model. Furthermore, benefiting from meta learning, our LMT tracker can work well after one iterative updating on the first frame. Encouraging results on OTB-2015, VOT-2016 and VOT-2018 demonstrate the effectiveness and robustness of our LMT tracker.

REFERENCES


BINENG ZHONG received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 2004, 2006, and 2010, respectively. From 2007 to 2008, he was a Research Fellow with the Institute of Automation and Institute of Computing Technology, Chinese Academy of Science. From September 2017 to September 2018, he is a visiting scholar in Northeastern University, Boston, MA, USA. Currently, he is an associate professor with the School of Computer Science and Technology, Huazhao University, China. His current research interests include pattern recognition, and computer vision.

GUORONG LI received the B.S. degree in technology of computer application from the Renmin University of China, Beijing, China, in 2006, and the Ph.D. degree in technology of computer application from the Graduate University of the Chinese Academy of Sciences, China, in 2012. She is currently an Associate Professor with the University of the Chinese Academy of Sciences. Her research interests include object tracking, video analysis, and cross-media analysis.

SICHENG ZHAO (SM’19) received the Ph.D. degree from Harbin Institute of Technology, Harbin, China, in 2016. He has been a Visiting Scholar at National University of Singapore from July 2013 to June 2014, and a Research Fellow at Tsinghua University from September 2016 to September 2017. He is currently a Research Fellow at University of California, Berkeley, USA. His research interests include affective computing, multimedia, and computer vision.

ZIYI CHEN received his Ph.D. degree in signal and information processing from Xiamen University, China, in 2016. He is currently a lecturer in the Department of Computer Science and Technology, Huazhao University, China. His current research interests include computer vision, machine learning, and remote sensing image processing.

WENTAO FAN received his MSc and PhD degrees in Electrical and Computer Engineering from Concordia University, Montreal, Quebec, Canada, in 2009 and 2014, respectively. He is currently an associate professor in the Department of Computer Science and Technology, Huazhao University, Xiamen, China. His research interests include machine learning, computer vision, and pattern recognition.

YITING LIN received the B.S. degrees in Information Management and Information Systems from the Beijing Technology and Business University, Beijing, China, in 2016. Currently, she is a Master student with the School of Computer Science and Technology, Huazhao University, Xiamen, China. Her current research interests include pattern recognition, machine learning, and computer vision.

WENTAO FAN received his MSc and PhD degrees in Electrical and Computer Engineering from Concordia University, Montreal, Quebec, Canada, in 2009 and 2014, respectively. He is currently an associate professor in the Department of Computer Science and Technology, Huazhao University, Xiamen, China. His research interests include machine learning, computer vision, and pattern recognition.