Object Tracking Using Spatio-Temporal Future Prediction

Yuan Liu, Ruoteng Li, Robby T. Tan, Member, IEEE, Yu Cheng, Xiubao Sui, Member, IEEE

Abstract—Occlusion is a long-standing problem that causes many modern tracking methods to be erroneous. In this paper, we address the occlusion problem by exploiting the current and future possible locations of the target object from its past trajectory. To achieve this, we introduce a learning-based tracking method that takes into account background motion modeling and trajectory prediction. Our trajectory prediction module predicts the target object’s locations in the current and future frames based on the object’s past trajectory. Since, in the input video, the target object’s trajectory is not only affected by the object motion but also the camera motion, our background motion module estimates the camera motion, so that the object’s trajectory can be made independent from it. To dynamically switch between the appearance-based tracker and the trajectory prediction, we employ a network that can assess how good a tracking prediction is, and we use the assessment scores to choose between the appearance-based tracker’s prediction and the trajectory-based prediction. Comprehensive evaluations show that the proposed method sets a new state-of-the-art performance on commonly used tracking benchmarks.

Index Terms—Object tracking, occlusion, trajectory prediction, background motion estimation.

1 INTRODUCTION

Visual object tracking is the task of estimating the location of a target object in each frame of an input video, while the initial location of the target is given in the first frame [1], [2], [3]. Recently, a Siamese learning paradigm [4], [5], [6], [7] has drawn attention in the field. These approaches cast the visual object tracking problem as learning a general similarity function by computing a cross-correlation between two image regions. Tracking is then performed by finding the feature representations learned for the search region most similar to those of the target-object template. As a result, the visual tracking problem can be solved by Siamese trackers using an end-to-end learning solution.

Despite their relative success, most current approaches including Siamese trackers tend to fail when occlusion occurs, and are erroneous when objects with similar appearance to the target object are present [8], [9]. We observe that most current trackers focus on developing discriminative target object’s feature representations [7], online appearance update mechanism [10] or re-detection [11]. A good feature representation or a good online update mechanism is important, however it can be still problematic when target is severely occluded. And, re-detection is just a partial remedy, i.e., useful only when the target object reappears from occlusion.

To address the occlusion problem, we aim at leveraging the predicted future trajectory. Unlike most existing methods, our basic idea is that, when the target object is severely/totally occluded, the future locations predicted from the motion trajectory are more reliable than the estimation of the appearance-based tracker. Thus, the future trajectory should be used instead of the tracker’s estimation. Based on this idea, we develop a trajectory-guided end-to-end network that predicts the target object’s possible locations in the future frames.

To realize the idea, the following are the insights of our algorithm. First, camera motion significantly affects the target object’s locations in the input frames, and should be taken into account. Second, once camera motion is compensated, the target object’s past trajectory represents the target object’s motion independent from the camera...
motion. Third, previous frames provide the appearance, motion, and surrounding information of the target object, which can contribute to reliable trajectory prediction.

Based on our insights, we design our deep network consisting of 4 subnetworks: an appearance-based tracking network (tracker), a background-motion network, a trajectory network, and an assessment network. The tracker estimates the target object’s locations solely from appearance information. It works robustly particularly when the appearance information in the input frame is sufficient and matches to the template. In our method, any appearance-based tracker from the state-of-the-art method can be used. The second network in our design, the background-motion network, estimates the camera motion. This camera motion is needed in our method, since the target object’s motion in the input video not only depends on the object motion in the world alone but also the camera motion. By estimating and then compensating the camera motion, we can obtain the target object’s trajectory independent from the camera motion. Our third network, the trajectory network, predicts the target object’s current and future locations from the target’s past observations.

Since our method produces two estimations for a single target object in one frame: one from the tracker and the other from the trajectory network, we need to choose one of them. For this, our fourth network, the assessment network, is trained to assess the accuracy of a prediction. As a result, we can have the confidence score for the tracker prediction and the confidence score for the trajectory prediction. We select the prediction that has higher confidence score. This allows our method to switch dynamically between the tracker and the trajectory prediction, depending on the severity of the occlusion. Note that, to make our confidence scores more reliable, we apply confidence score calibration [12]. Figure 1 shows our tracking results even when the target objects are totally occluded.

The summary of our contributions is as follows:

- We introduce a background-motion network that captures the global background motion between consecutive frames to represent the camera motion. This background motion is important to make the target object motion trajectory independent from camera motion.
- We propose a trajectory network that learns from the target object’s observations in several previous frames and predicts the locations of the target object in the subsequent future frames. A multi-stream conv-LSTM architecture is introduced to encode and decode the temporal changes of the target object locations in the past.
- We present a unified tracking model that integrates background motion modeling, appearance-based tracking, and trajectory prediction to solve the occlusion problem. This unified tracking model can be trained end-to-end.
- We introduce a tracking selection mechanism based on the tracker and trajectory confidence scores, which enables our tracking method to switch automatically and dynamically between the appearance-based tracker and our trajectory-based prediction, particularly when occlusion occurs.

The large part of this paper has been published in ECCV 2020 [13]. Compared with our ECCV 2020 paper, new contents in this paper includes: unifying the background motion modeling, tracking and trajectory prediction into one deep network (Section 3). This unified model can be trained end-to-end, which further enhances the performance of our trajectory prediction. Second, to handle overconfident scores of the trajectory confidence score, we employ confidence calibration in the trajectory-guided tracking mechanism (Section 3.4), which can lead to a significant performance gain. Moreover, we provide the details of our implementation, as well as more comprehensive analysis and evaluation.

2 RELATED WORKS

Visual Object Tracking Visual tracking has witnessed a rapid boost in recent years, with the development of a variety of approaches. Recently, the approaches based on Siamese networks [4], [5], [6], [7] have received significant attentions due to their well-balanced tracking accuracy and efficiency. Instead of learning a discriminative classifier online, these approaches formulate visual tracking as a cross-correlation problem and can easily be trained end-to-end using pairs of annotated images. Based on this idea, a number of methods enhance the tracking performance by making use of region proposals [14], hard negative mining [7] and binary segmentation [6]. A few recent works [15], [16], [17], [18], [19] aim to employ temporal information for better object feature representation. Yang et al. in [15] estimate an object-specific filter for tracking by using a Recurrent Neural Networks(RNN). Cui et al. in [16] propose a multi-directional RNN to capture long-range contextual cues. Yang et al. in [17] propose a dynamic memory network to update the feature representation model. [18] utilizes a gated recurrent unit to model the temporal information of sequences, while [19] embeds an attention mechanism into the RNN to help searching target.

Most modern trackers focus on modelling the object feature representation to improve tracking performance. However, relying only on object feature representation can be problematic in the case where the target is occluded or the target comes across other objects with similar appearance. A powerful model update strategy is an effective way to discriminate the similar objects from the target or to handle some partial occlusion cases. However, occlusion is still challenging for these methods. To handle occlusion, some trackers utilize the failure re-detection strategy to search target when the target object appears again after occlusion. However, failure re-detection cannot help to track the target during occlusion period. To address the occlusion problem completely, we combine the object representation in the spatial domain with the trajectory prediction in the temporal domain by using a spatio-temporal network to track target accurately and robustly.

Trajectory Prediction Unlike the visual object tracking task, trajectory prediction aims at predicting the target’s possible positions in future frames [20]. Recently, a large body of works focus on person trajectory prediction by considering
human social interactions and behaviors in crowded scenes. Zou et al. in [21] learn human behaviors in crowds by using a decision-making process. Liang et al. in [22] utilize rich visual features about human behavioral information and interaction with their surroundings to predict future path jointly with future activities. A number of methods try to learn the effects of the physical scene. Scene-LSTM [23] divides the static scene into Manhattan grid and predict pedestrian’s location using LSTM. SoPhie [24] combines deep-net features from a scene semantic segmentation model and generative adversarial network using attention to model person trajectories.

There are some methods that take the motion prediction into account for tracking or predicting person path. Amir et al. in [25] propose a structure of RNN that jointly reasons on multiple cues over a temporal window for multi-target tracking. Ellis et al. in [26] propose a Gaussian process regression model for pedestrian motion. Hogg et al. in [27] propose a statistical model of object trajectories which is learned from image sequences. Compared with these methods, which assume a static camera in modeling the trajectory, our idea is to integrate trajectory prediction into object tracking task using deep learning for a dynamic camera. To simplify the trajectory complexity, several methods split motion into camera motion and object motion. In particular, Takuma et al. in [28] recently propose an accurate method that makes use of camera ego-motion, scales, the speed of the target person, and the person pose to predict person’s location in future frames. [29] proposes an end-to-end trajectory prediction method using an LSTM-CNN network that models the interactions between road-agents in dense traffic. Unlike these methods that make full use of the object surroundings, which are expensive for general single object tracking, we utilize only the past trajectory and target visual features to predict short-term future locations for assisting the tracker to handle occlusion.

3 Proposed Method

We develop a tracking method that addresses the occlusion problem. As shown in Figure 2, our network consists of 4 subnetworks: an appearance-based tracking network (tracker), a background-motion network, a trajectory network, and an assessment network. The tracking network (tracker) estimates the target location \( l_{\text{now}} \) from appearance inference. The background-motion network captures the global background motion vectors \( \{ m_{\text{bg}}^{\text{prev}}, \ldots, m_{\text{bg}}^{\text{now}-1} \} \) to compensate target's trajectory. The trajectory network predicts target’s current and future locations \( \{ l_{\text{now}}^{\text{traj}}, \ldots, l_{\text{future}}^{\text{traj}} \} \) from its past observations. The assessment network assess the confidence scores \( s_{\text{now}}^{\text{tracker}} \) and \( s_{\text{now}}^{\text{traj}} \) of \( l_{\text{now}}^{\text{traj}} \) and \( l_{\text{now}}^{\text{tracker}} \), respectively. The final result \( l_{\text{now}}^{\text{proj}} \) is selected by comparing the calibrated confidence scores.

![Fig. 2. Overview of our tracking method. Our approach consists of 4 networks: tracking network (tracker), a background-motion network, a trajectory network, and an assessment network. The tracking network (tracker) estimates the target location \( l_{\text{now}}^{\text{tracker}} \) from appearance inference. The background-motion network captures the global background motion vectors \( \{ m_{\text{bg}}^{\text{prev}}, \ldots, m_{\text{bg}}^{\text{now}-1} \} \) to compensate target’s trajectory. The trajectory network predicts target’s current and future locations \( \{ l_{\text{now}}^{\text{traj}}, \ldots, l_{\text{future}}^{\text{traj}} \} \) from its past observations. The assessment network assess the confidence scores \( s_{\text{now}}^{\text{tracker}} \) and \( s_{\text{now}}^{\text{traj}} \) of \( l_{\text{now}}^{\text{traj}} \) and \( l_{\text{now}}^{\text{tracker}} \), respectively. The final result \( l_{\text{now}}^{\text{proj}} \) is selected by comparing the calibrated confidence scores.](image-url)
3.2 Background Motion Network

Predicting \( L^{\text{future}} \) directly from the tracker’s estimation \( L^{\text{prev}} \) is problematic due to significant camera motion present in the input video. More specifically, the coordinate system representing each location \( t' \) changes dynamically as the camera moves; since the target object’s motion appearing on the input sequence depends on both the object motion in the world and the camera motion. To address the problem, the camera motion should be estimated and compensated, so that the object motion and thus the object trajectory can be independent from the camera motion.

Generally, the camera motion between adjacent frames, can be represented by rotation and translation. Rotation is described by rotation matrix \( R^t \in \mathbb{R}^{3 \times 3} \) and translation is described by vector \( V^t \in \mathbb{R}^3 \), both from a frame at \( t - 1 \) to a frame at \( t \). To efficiently estimate the camera motion parameters, we cast the camera motion estimation as matching the global background between two consecutive frames. Specifically, the 3-D rotation matrix \( R^t \) is simplified to one rotation angle \( r_t \in \mathbb{R}^{2 \times 2} \) in the 2-D image domain, and the translation vector \( V_t \) is split to the translation vector \( v_t \in \mathbb{R}^2 \), and scale changing, \( c' \).

To detect the global background motion, we employ a Siamese network that compares the adjacent frames as
shown in Figure 4. As the Siamese network is a robust method for image matching, it also can be used to match global background motion. Unlike the Siamese network for the tracker, the Siamese network here compares the similarity of the global background scenes instead of the target object. Thus, the patch \( z_{t-1} \) is cropped at the center of the region in the frame \( t-1 \), and the search image \( x^t \) is the larger cropped image patch in the frame \( t \). To avoid the interference of the target object, the region is masked with one value, i.e., the average value of the whole image region. The heatmap \( G_{bg} \) of background matching in the frame \( t \) can be produced by applying:

\[
G^t_{bg} = f_{bg}(x^t_{now} \ast f_{bg}(z_{t-1}^t), \tag{2}
\]

where \( f_{bg} \) represents the background-motion network.

To be able to handle multiple scales and rotations, we search image patches \( x_{c1}, ..., x_{cn} \) and \( x_{r1}, ..., x_{rn} \), by using patch pyramid with different scale factors \( c1, ..., cn \) and rotation factors \( r1, ..., rn \). The background motion translation vector \( v^t_{now} \) is achieved by finding the displacement of the maximum value and the center of the heatmap. The scale change \( c^t_{now} \) and the rotation change \( r^t_{now} \) are estimated by searching the position of the maximum value in the multi-scale heatmap set and the multi-rotation heatmap set, respectively.

To represent the global motion along multiple frames, we accumulate these vectors, which represent the local movement between two adjacent frames. Therefore, for each frame within the input interval \([t_{prev}, t_{now} - 1]\), we apply:

\[
m^t_{bg} = \begin{cases} 
  v^t_{bg} & (t = t_{prev} + 1) \\
  r^t_{bg} + m^t_{bg} & (t > t_{prev} + 1)
\end{cases}, \tag{3}
\]

where \( m^t_{bg} \) denote the background motion patterns of frame \( t \) in the global background coordinate system. \( t \in [t_{prev}, t_{now} - 1] \). \( v^t \), \( r^t \) and \( c^t \) denotes the translation, rotation and scale change between two adjacent frames at \( t - 1 \) and \( t \).

As shown in Figure 5, having compensated the background/camera motion, the trajectories follow the target objects independently from the camera motion. It also shows the Siamese method is able to match global background robustly, even when some moving foreground objects are present.

### 3.3 Trajectory Network

The appearance-based tracker is erroneous when the appearance of the target object is degraded/occluded severely. To address this problem, we rely on the target object’s past observations to predict the current most possible location, which also implies predicting a future location. The straightforward way to predict future locations of the target object is to utilize its previous immediate location (i.e., the temporal smoothness constraint). However, it is insufficient for the object tracking task, since the target object’s motion can be arbitrary and the occlusion occurs in a few consecutive frames.

To predict future locations, we develop a fully-convolutional deep network that utilizes a multi-stream conv-LSTM architecture shown in Figure 5 which is expressed as:

\[
P^t = d(y^t_{lstm}), \tag{4}
\]

where:

\[
\begin{align*}
  t & \in [t_{now}, t_{future}] \\
  y^t_{lstm} & = LSTM(x^t_{lstm}, y^{t-1}_{lstm}) \\
  x^t_{lstm} & = concat(f_{\mu}(H^k), f_{\theta}(I^k)) \\
  k & \in [t_{prev}, t_{now} - 1]
\end{align*}
\]

\( P^t \) is the response map of a frame at time \( t \). Operator \( concat \) represents the concatenation operation of the feature maps. \( LSTM \) indicates conv-LSTM. \( y^t_{lstm} \) denotes the output of the LSTM cell at frame \( k \). \( f_{\mu} \) and \( f_{\theta} \) represent two convolutional networks with network parameters \( \mu \) and \( \theta \), respectively. \( d \) denotes the deconvolutional network. \( I^k \) is the input image at frame \( k \), and \( H^k \) is its location map. We obtain \( H^k \) by generating a Gaussian function on a map where the peak is at \( t^k \); the size of the map is the same as the size of the input frame.

Let us denote the annotated ground-truth location of the target object \( GT^t \) on a frame at time \( t \). The loss function for the trajectory prediction is an L1-loss over all predicted future frames:

\[
L_{traj} = \sum_{t=t_{now}}^{t_{future}} ||P^t - N(GT^t)||, \tag{5}
\]

where \( N \) indicates a Gaussian function, which the mean is at \( GT^t \). Consequently, we can obtain the trajectory prediction, \( l^{t_{now}}_{traj} \) from the peak of \( P^t_{now} \). The same applies to the future predictions, \( l^{t_{now}+1}_{traj}, ..., l^{t_{future}}_{traj} \).

### 3.4 Assessment Network

Up to this stage, for one frame and a single target object, there are two predicted locations in our method: One is from the tracker model, \( l^{t_{now}}_{tracker} \), and the other is from the trajectory module, \( l^{t_{now}}_{traj} \). Hence, we need to select them so that we can have one predicted location, \( l^{t_{now}}_{pred} \).To handle this problem, we create a network that, given the location prediction, can assess how correct a tracking prediction is.

The network can be expressed as:

\[
s^{t_{now}}_{\ast} = s(l^{t_{now}}_{\ast}, L^{pre}_{pred} + m^{pre}_{bg}, G^{now}_{tracker}), \tag{6}
\]

where \( s^{t_{now}}_{\ast} \) is the confidence score of how good the tracking prediction, \( l^{t_{now}}_{\ast} \), is. Function \( s \) represents the assessment network. \( L^{pre} = \{l^{t_{prev}}, ..., l^{t-1}\} \) and \( m^{pre}_{bg} = \{m^{t_{prev}}_{bg}, ..., m^{t-1}_{bg}\} \).

This equation also implies that, given the information \( \{L^{pre}_{pred}, m^{pre}_{bg}, G^{now}_{tracker}\} \), the network predicts how correct the trajectory prediction \( l^{t_{now}}_{\ast} \) is.

Note that, \( l^{t_{now}}_{now} \) can represent either \( l^{t_{now}}_{tracker} \) or \( l^{t_{now}}_{\ast} \), implying the network can be used to assess the prediction from the tracker and the trajectory prediction, generating two scores: \( s^{t_{now}}_{\ast} \) and \( s^{t_{now}}_{\ast} \). Once we obtained \( s^{t_{now}}_{\ast} \) and \( s^{t_{now}}_{\ast} \), then we can compare them, and then select which tracking prediction is more reliable. This will enable our method to switch automatically and dynamically between the tracker and trajectory predictions.

To train the assessment network, \( s \), we rely on a binary cross entropy:

\[
L_s = a \log(s^{t_{now}}_{\ast}) + (1 - a) \log(1 - s^{t_{now}}_{\ast}), \tag{7}
\]
where \( a \in \{1, 0\} \) is the ground-truth. \( a = 1 \) means the prediction is correct, and \( a = 0 \) means the prediction is wrong. The network is trained on positive and negative samples. By computing the intersection over union (IoU) between the ground truth box and the bounding box where the center locates \( I_{\text{now}} \), this sample will be considered as a positive sample; specifically, when the IoU is larger than 0.5. The corresponding negative sample is generated by adding a random drift displacement on the \( I_{\text{now}} \) of the positive samples.

**Trajectory Confidence Calibration** Since we rely on the confidence scores to decide the tracking final prediction, we must ensure that the confidence scores are consistent and linear to the prediction accuracy. For this reason, utilizing a calibration technique [12], we calibrate our confidence scores.

Let \( s^t_{\text{raw}} \) is the non-calibrated score of the assessment network \( s \). The calibrated score is obtained by applying:

\[
s^t_s = \max_i \sigma \left( s^t_{\text{raw}}(i)/T \right).
\]

In our case, \( k = 2 \). \( T \) is the temperature, \( T > 0 \), which is optimized with respect to negative log likelihood (NLL) [20], [31] on the validation set. The network’s parameters are fixed during this calibration process.

## 4 IMPLEMENTATION DETAILS

### 4.1 Overall Algorithm

Algorithm 1 summarizes our overall method. Given the input frames and previous target locations, we want to estimate the current location, as well as the future locations. To achieve this, at step 1, we compute the heatmap, \( G_{\text{tracker}} \), from Eq. (1). In this equation, we require the object template, \( z^t \), which can be obtained from the previous tracking prediction, to search the target object in the current frame. In step 2, we obtain the target object’s location, \( l^t_{\text{now}} \), from the maximum value in map \( G^t_{\text{now}} \).

In step 3, we compute the background-motion heatmaps \( \{ G^t_{bg}^{prev}, ..., G^t_{bg}^{now-1} \} \) using Eq. (3). To apply this matching equation, we need template \( z^t \) and search region \( x^t \) of the respective frames. In step 4, we can obtain \( \{ m^t_{bg}^{prev}, ..., m^t_{bg}^{now-1} \} \), which are the background motions from frame \( I_{\text{prev}} \) to frame \( I_{\text{now}} \).

Once we have the background motion information, we can predict the trajectory. In step 5, we generate the location maps, \( \{ H^t_{\text{prev}}, ..., H^t_{\text{now}} \} \), which each of them is represented by a map containing a Gaussian function whose peak is at the predicted location. These maps are independent from the background/camera motion. In step 6, we compute the response maps, \( \{ P^t_{\text{now}}, ..., P^t_{\text{future}} \} \), using our conv-LSTM networks described in Eq. (4). Based on the current frame response map, \( P^t_{\text{now}} \), we can obtain the predicted location of the target object, \( l^t_{\text{traj}} \).

Hence, we have two predictions: \( l^t_{\text{now}} \) from step 7 and \( l^t_{\text{traj}} \) from step 7. In step 8, based on Eq. (6), we compute the confidence scores, and in step 9, we calibrate them, producing \( s^t_{\text{now}} \) and \( s^t_{\text{traj}} \). Based on the comparison between these confidence scores, in step 10, we choose either \( l^t_{\text{now}} \) or \( l^t_{\text{traj}} \) as our final tracking prediction for the current frame.

### 4.2 Network Architecture

Figure 6 shows the pipelines of the training and testing of our implementation. We use the target object’s observations
in the previous 11 frames to predict the locations of the target in next 5 future frames. For the trajectory network, we use a general seq-to-seq LSTM network [32] as our backbone, where the encoder consists of 11 LSTM cells and the decoder consists of 5 LSTM cells. As the input sequence is a spatio-temporal data, we extend the matrix multiplications of LSTMs to a convolution operation on the multi-channel feature maps, which preserves the spatial information of the target object.

The size of all convolutional filters is $3 \times 3$. A two stream of fully-convolutional network with 5 convolutional layers of stride 4 extracts features from the location map and image. The two sets of feature maps are concatenated before the LSTM network. After decoding of the LSTM, each response map representing the future location state is generated by three deconvolutional layers with stride 4. Finally, based on the target object’s past 11 locations and the tracker’s current heatmap score, the confidence score is obtained from a network of 3 1-D convolutional layers followed by a sigmoid activate function. For the background motion prediction network, we choose the popular Siamese network Siamfc [4]. For our tracking network, we choose an existing tracking method’s network (ie, DiMP [10] or SiamMask [5]).

### 4.3 Training

For the trajectory network, we choose 16 consecutive frames in the input video as the temporal range of one sample set. For each sample set, we select the first 11 frames as the past states and regard the rest 5 frames as the future states. For the trajectory classifier network, we choose the target object’s locations in first 12 frames from the successive 16 frames as one sample set. For each sample set, the last frame location is the future state. Specifically, we consider data augmentation by random translations (up to the mean of the displacements in the past frames).

During training, the background-motion network and the tracking network use the pre-trained models in Figure 6, which means the parameters of these two networks are fixed. Thus, the input of our unified model is 12 frames sequence (11 previous frames and current frame), while the output is 5 response maps in future 5 frames and a confidence score that represents the trajectory confidence of the trajectory prediction status. Note that, the assessment network only outputs one score, and performs twice as mentioned in Section 3.4. The label of the response map is produced by a Gaussian function. The peak location is based on the target object’s location by compensating the background motion vector. We use the Adam optimization with the fixed learning rate of 0.0002 for 20 epochs. We train our unified model by using ImageNet-VID [33].

### 4.4 Testing

During testing, the background-motion network and trajectory network take in 11 previous frames and the target’s past locations to predict 5 future locations. The tracking network only process the current frame to estimate the target location and also provide the heatmap. The assessment network outputs the confidence score of the trajectory prediction and tracking status. Note that, our method needs the first 11 frames states. To deal with the initialization problem, we repeat the first frame state to reach the 11 frames initialization quantity at the first 11 frames. It can be seen as the target object stays still at the initial location. Our trajectory obtains the motion information from the tracker’s results. For faster processing speed, it is not necessary to predict trajectory every frame until the tracker gets a low confidence score.

### 5 EXPERIMENT RESULTS

We evaluate our approach on VOT-2019 [42], OTB2013 [1], OTB2015 [2] and UAV123 [43] benchmarks. We choose the Siamese framework based tracker DIMP [10] and SiamMask [5] as our baseline trackers. On a single Nvidia GTX 1080Ti GPU, we achieve a tracking speed of 11 FPS when employing DIMP as the base tracker and 13 FPS for SiamMask.

In this section, we provide a comprehensive comparison of Ours-DIMP with 12 state-of-the-art methods in the literature. The trackers used for our comparison are: DaSiamRPN [7], SiamRPN [6], ATOM [36], CCOT [38], MDNet [39], ECO [40], SiamRPN++ [34], UPDT [41], DCFST [35], DeepSTRCF [37], SiamMask [5] and DIMP [10]. Some of these trackers use the Siamese framework as backbone.

#### 5.1 Evaluation Results

**VOT2019 Dataset [42]**: We evaluate our approach on the 2019 version of Visual Object Tracking (VOT) consisting of 60 challenging videos. Following the evaluation protocol of VOT2019, we adopt the expected average overlap (EAO), accuracy (average overlap over successfully tracked frames) and robustness (failure rate) to compare different trackers. The detailed comparisons are reported in Table 1. Compared to DIMP, our approach has a 6% lower failure rate, while achieving a better accuracy. This shows that trajectory prediction is crucial for robust tracking.
TABLE 1
State-of-the-art comparison on the VOT2019 dataset in terms of expected average overlap (EAO), accuracy and robustness. Compared with the baseline tracker DIMP, our approach Ours-DIMP achieves improvement on all three metrics and obtains a significant performance gain on the robustness.

|       | SiamRPN++ [34] | DCFST [35] | ATOM [36] | SiameseMask [5] | DIMP [10] | Ours-DIMP |
|-------|----------------|------------|-----------|-----------------|-----------|-----------|
| EAO   | 0.285          | 0.361      | 0.292     | 0.287           | 0.305     | 0.323     |
| Accuracy | 0.599      | 0.589      | 0.603     | 0.602           | 0.589     | 0.596     |
| Robustness | 0.482    | 0.321      | 0.411     | 0.426           | 0.361     | 0.301     |

TABLE 2
State-of-the-art comparison on OTB2013, OTB2015 and UAV123 datasets in terms of area-under-the-curve (AUC) score. The best results are shown in bold. Ours-DIMP improves the base tracker DIMP on all three datasets. Especially, our approach achieves the best AUC score on the OTB2015 and UAV123 datasets.

|       | DeepSTRCF [37] | DaSiamRPN [7] | ATOM [36] | CCOT [38] | MDNet [50] | ECO [40] | SiamRPN++ [34] | UPDT [41] | DiMP [10] | Ours-DIMP |
|-------|----------------|---------------|-----------|-----------|------------|----------|----------------|-----------|-----------|-----------|
| OTB2013 | 68.2          | 66.0          | 66.0      | 68.0      | 71.1       | 71.2     | 69.1           | 71.4      | 69.1      | 69.9      |
| OTB2015 | 67.5          | 66.0          | 66.7      | 67.3      | 67.8       | 69.1     | 69.6           | 69.1      | 67.7      | 70.1      |
| UAV123  | -              | 56.9          | 64.3      | 50.7      | -          | 50.6     | 64.2           | 54.2      | 64.3      | 65.1      |

OTB2013 Dataset [1]: This dataset consists of 50 videos with different attributes including occlusion. Table 2 shows the AUC scores on the OTB2013 dataset. Among the compared methods, UPDT [41] achieves the best results with an AUC score of 71.4%. Ours-DiMP achieves an AUC score of 69.9%, compared with our baseline tracker DiMP 69.1%. We also report overlap success plots on the OTB2013 in Figure 7.

OTB2015 Dataset [9]: Table 2 shows the AUC scores over all the 100 videos in the dataset. Ours-DiMP achieves the best results with an AUC score of 70.1%, compared with our baseline tracker DiMP 67.7%. We also report overlap success plots on the OTB2015 in Figure 7. Our approach improves the baseline tracker DiMP on both OTB2013 and OTB2015. Compared with our little improvement on OTB2013, our approach obtains a significant performance gain on OTB2015, which contains more occlusion videos.

UAV123 Dataset [43]: This dataset consists of 123 low altitude aerial videos captured from a UAV. Compared to other datasets, UAV123 has heavier camera motion that affects the target’s trajectory severely. Testing on UAV dataset is challenging for our approach based on trajectory prediction. AUC results are shown in Table 2. Ours-DiMP achieves the best AUC score of 65.1%, verifying the strong trajectory prediction abilities of our tracker under the heavy camera motion.

5.2 Attribute-based Analysis
To analyze the performance on occlusion and other video attributes, we compare our method with the baseline tracker DiMP on OTB2015 and UAV123 datasets. In the OTB and UAV datasets, all videos are annotated with 11 different attributes, namely: low resolution, background clutters, out-of-view, occlusion, motion blur, scale variation, deformation, illumination variation, fast motion, out-of-plane rotation and in-plane rotation. Table 3 shows the AUC scores of all the 11 attributes in the OTB2015 dataset. Our approach provides favorable results on all 11 attributes compared with DiMP. Specially, our method achieves a significant gain of 4.2% on the occlusion attribute and obtains a improvement of about 4% in AUC score on low resolution, background clutter, out of view attributes. Our method achieves the best AUC score on occlusion attribute in Figure 8. The AUC scores of the 11 attributes in the UAV123 dataset are reported in Table 4. Our method outperforms DiMP with a relative gain of 1.6% on the occlusion attribute. Our method also achieves the best AUC score on occlusion attribute in Figure 9.

Since the occlusion and out of view attributes diminish the target object’s appearance in the image frame, it is...
TABLE 3
Attribute-based comparison with state-of-the-art trackers on the OTB2015 dataset. We report the AUC scores (%) for the state-of-the-art trackers. The number of videos associated with the attribute is shown in parenthesis. Compared with DIMP, our approach provides improved performance on all 11 attributes, especially occlusion.

| Attribute                  | Ours-DIMP | DIMP [10] |
|----------------------------|-----------|-----------|
| Low resolution             | 63.3      | 64.6      |
| Background clutter         | 67.5      | 63.5      |
| Out of View                | 64.6      | 63.3      |
| Occlusion                  | 68.2      | 69.0      |
| Motion blur                | 74.9      | 68.0      |
| Scale variation (65)       | 70.6      | 68.8      |
| Deformation                | 70.9      | 70.9      |
| Illumination variation (38)| 69.1      | 66.5      |
| Fast motion (42)           | 67.9      | 67.7      |
| Out-of-plane rotation (63) | 69.0      | 68.5      |
| In-plane rotation (51)     |           |           |

TABLE 4
Attribute-based comparison with the baseline tracker on the UAV123 dataset. We report the AUC scores (%). The number of videos associated with the attribute is shown in parenthesis. Compared with DIMP, our approach provides improved performance on 8 of 11 attributes, especially occlusion.

| Attribute                  | Ours-DIMP | DIMP [10] |
|----------------------------|-----------|-----------|
| Low resolution             | 66.7      | 64.9      |
| Background clutter         | 64.0      | 62.2      |
| Out of View                | 62.0      | 60.8      |
| Occlusion                  | 57.5      | 49.5      |
| Motion blur                |           | 43.5      |
| Scale variation (48)       |           | 62.6      |
| Deformation                |           | 58.4      |
| Illumination variation (109)|         | 61.6      |
| Fast motion (30)           |           |           |
| Out-of-plane rotation (68) |           |           |
| In-plane rotation (21)     |           |           |

5.3 Ablation Studies
We perform an analysis on the proposed model prediction architecture. Experiments are performed on VOT2018 [8] dataset.

Impact of Different Baseline Trackers Since the baseline tracker provides our method the heatmap score and tracking state, its performance is important to our method. To analyze the influence of baseline tracker, we choose another popular Siamese-framework based tracker SiamMask [5] as our baseline tracker. Compared with SiamMask, our method, namely Ours-SiamMask, improves the performance on all EAO, accuracy and robustness criteria. In particular, Ours-SiamMask obtains a significant relative gain of 5.7% in EAO, compared to SiamMask. Compared with the baseline variant tracker SiamMask-LD which improved by training on larger dataset, our corresponding, namely Ours-SiamMask-LD, achieves the gain of 2% and 8.4% in EAO and robustness respectively. Our method obtains a processing speed of 13 FPS which includes the processing time of the baseline tracker. The results for this analysis verifies the strong generalization abilities of our method, as shown in Table 5.

Impact of Multi-Cues We make an ablation study to see how the background motion cue, the location map cue and image cue contribute overall tracking performances respectively. We compare three different inputs. No bg: The trajectory prediction network predicts the target object’s locations without background motion compensation. Thus, the camera motion will affects the target object’s trajectory. No img: Here, we use only the location map cue and the background motion cue. No loc: We utilize the target object’s location value directly instead of the location map. The results are shown in Table 6. No bg achieves an EAO score of 0.394, even worse than the baseline tracker. No
Impact of Temporal Range
We analyze the impact of the input’s temporal range. Our basic idea is that using a short-term time window of one or two seconds for observation to predict the target object’s future location. Thus, we choose 21 frames and 51 frames as our temporal ranges, namely Temp-21 and Temp-51 respectively. Our complete method, namely Ours, takes 11 frames, as mentioned in section 4. The results are reported in Table 7. The Temp-21 variant outperforms No loc by 0.7%. This shows that the location map is a better way to represent the target object’s trajectory in the image domain. Our complete method obtains the best results, which means each cue in our inputs is beneficial to improve performance.

Impact of Selection Mechanism
We analyze the impact of trajectory selection mechanism by comparing three different variants. No heatmap: the selection score is evaluated based on the target trajectory without the heatmap score. Weight: The heatmap score is treated as a weight to the confidence score instead of the trajectory classifier network’s input. Ours: A sub-classifier network takes in target’s locations and corresponding heatmap score, and outputs a confidence score, as described in section 3.4. The results are shown in Table 8. No heatmap even makes the baseline tracker worse. In contrast, by considering the heatmap score from the tracker’s appearance inference, our method obtains a significant gain of about 2% in EAO score over the baseline tracker. It also indicates that combing the heatmap score into a network is a more effective way than using it as a weight. These results demonstrate that our method can effectively switches between the tracker state and trajectory prediction.

Impact of Confidence Calibration and Unify Model
We analyze the impact of Confidence Calibration and unified model by comparing three different variants. No Calibration: the final result is selected by using trajectory confidence score without calibration. No Unify: The background motion modeling, appearance-based tracking, and trajectory prediction are trained and tested independently. Ours: Our complete method unifies the model and calibrates the trajectory confidence. The results are shown in Table 9. Our complete method obtains the best results, which means calibrating confidence and unifying model are beneficial to improve performance. Furthermore, a qualitative comparison example is shown in Figure 10. In the left image in Figure 10, when the baseline tracker tracks drift due to the similar appearance object nearby, our approach assistant tracker to find the right target object by comparing their trajectory confidence. However, when the wrong tracking result is close to the target’s past trajectory in the middle image of Figure 10, the trajectory confidence is confused to classify. By calibrating confidence, it can be solved in the right image of Figure 10.
Fig. 11. Qualitative results of Ours-SiamMask for sequences from VOT2018. The sequence names, from top to bottom, are soccer1, girl, conduction1, and basketball. The red dots denote the target’s past locations and the blue ones are our prediction. In comparison with the ground truth (green bounding box), our method (red one) performs well under full occlusion in Girl and Soccer1 sequences.

### TABLE 9

| Tracker          | No calibration | No unify | Ours  |
|------------------|----------------|---------|-------|
| EAO ↑            | 0.422          | 0.437   | 0.440 | 0.442 |

### 5.4 Qualitative Evaluation

Here we provide a qualitative comparison of our approach. Figure 11 illustrates frames from four sequences with occlusion (girl and soccer) and significant camera motion (girl, soccer, conduction1 and basketball). In the girl sequence, the target girl is totally hidden in occlusion. It is difficult to find any useful visual information in the image. By predicting future locations from the target’s past trajectory, our approach prevents the tracker drift. In the soccer sequence, the background is clutter and the target is partly occluded. In the conduction1 sequence, the unobvious appearance of the target is challenging for the general appearance-based tracker. In the basketball sequence, it is also difficult for tracking due to the target motion changing dramatically with surrounding and the similar objects nearby. These cases can be solved by our approach using trajectory prediction in Figure 11. All these four sequences contain significant camera motion, our approach performs well compared with ground truth, while the trajectory is also shown smooth in Figure 11. This shows the strong compensating trajectory ability of our background-motion estimation. Again, our approach demonstrates robustness in these scenarios and is able to keep tracking the target throughout the sequence.

### 6 Conclusions

We introduce a novel tracking method, which is trained in end-to-end, and combines background motion modeling, appearance-based tracking, and trajectory prediction to solve the occlusion problem. We present a background motion model that captures the global background motion between adjacent frames to the effect of camera motion. We propose a new trajectory prediction model that learns from the target object’s observations in several previous frames and predicts the locations of the target object in the subsequent future frames. A multi-stream conv-LSTM architecture is introduced to encode and decode temporal evolution in these observations. We also present a trajectory-guided tracking mechanism by using a calibrated trajectory confidence score, which helps the tracker to switch dynamically between the current tracker prediction and our trajectory prediction, particularly when the target object is occluded.

We conduct extensive experiments on the Online Tracking Benchmark (OTB), the Unmanned Aerial Vehicle (UAV) and the Visual Object Tracking (VOT) 2019, 2018 challenge datasets. The results clearly demonstrate that our approach provides significant improvement over the baseline trackers DIMP [10] and SiamMask [5], especially under occlusion. We also compare our approach with several state-of-the-art trackers. Our method outperforms 8 state-of-the-art trackers in the literature on the OTB and UAV datasets. For the occlusion attribute on the OTB
and UAV datasets, our tracking approach is shown to outperform 8 state-of-the-art trackers, obtaining the top ranking AUC score.

ACKNOWLEDGMENTS

Robby T. Tan’s research in this work is supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative.

REFERENCES

[1] A. W. Smeluders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 7, pp. 1442–1468, 2013.

[2] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” ACM computing surveys (CSUR), vol. 38, no. 4, p. 13, 2006.

[3] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, “Recent advances and trends in visual tracking: A review,” Neurocomputing, vol. 74, no. 18, pp. 3823–3831, 2011.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in European conference on computer vision. Springer, 2016, pp. 850–865.

[5] Q. Wang, L. Zang, L. Bertinetto, W. Hu, and P. H. Torr, “Fast online object tracking and segmentation: A unifying approach”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1328–1338.

[6] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with siamese region proposal network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8971–8980.

[7] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, “Distractor-aware siamese networks for visual object tracking,” in Proceedings of the European conference on Computer Vision (ECCV), 2018, pp. 101–117.

[8] M. Cristan, A. Leonidas, J. Matas, M. Felsberg, R. Pfugfelder, L. Cehovin Zajc, T. Vojir, G. Bhat, A. Lucezic, E. Eldeesokey et al., “The sixth visual object tracking vot2018 challenge results,” in Proceedings of the European conference on Computer Vision (ECCV), 2018, pp. 0–0.

[9] Y. Wu, J. Lim, and M.-H. Yang, “Object tracking benchmark,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1834–1848, 2015.

[10] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, “Learning discriminative model prediction for tracking,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6182–6191.

[11] Y. Zheng, M. Tang, H. Lu, “Reliable re-detection for long-term tracking,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 3, pp. 730–743, 2018.

[12] C. Guo, C. Pleiss, K. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” arXiv preprint arXiv:1706.04599, 2017.

[13] V. Andrea, B. Horst, B. Thomas, and F. Jan-Michael, Computer Vision–ECCV 2020. Springer International Publishing, 2020, vol. 12348, no. 1.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[15] T. Yang and A. B. Chan, “Recurrent filter learning for visual tracking,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2010–2019.

[16] M. Zilu, S. Xiao, J. Feng, and S. Yan, “Recursively target-attending tracking,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1449–1458.

[17] T. Yang and A. B. Chan, “Learning dynamic memory networks for object tracking,” in Proceedings of the European conference on Computer Vision (ECCV), 2018, pp. 152–167.

[18] Q. Gan, Q. Guo, Z. Zhang, and K. Cho, “First step toward model-free, anonymous object tracking with recurrent neural networks,” arXiv preprint arXiv:1811.06425, 2018.

[19] S. Ebrahimii Kahou, V. Michalski, R. Memisevic, C. Pal, and P. Vincent, “Ratm: Recurrent attentive tracking model,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 10–19.

[20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social ISTM: Human trajectory prediction in crowded spaces,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 961–971.

[21] H. Zou, H. Su, S. Song, and J. Zhu, “Understanding human behaviors in crowds by imitating the decision-making process,” in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[22] J. Liang, L. Jiang, J. C. Niebles, A. G. Hauptmann, and L. Fei-Fei, “Peeking into the future: Predicting future person activities and locations in videos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5725–5734.

[23] H. Manh and G. Alahbainh, “Scene- lstm: A model for human trajectory prediction,” arXiv preprint arXiv:1808.04018, 2018.

[24] A. Sadeghian, A. Alahi, and S. Savarese, “Tracking the untrackable: Learning to track multiple cues with long-term dependencies,” in Proceedings of the IEEE Conference on Computer Vision, 2017, pp. 300–311.

[25] D. Ellis, E. Sommerladre, and I. Reid, “Modelling pedestrian trajectory patterns with gaussian processes,” in 2009 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, 2009, pp. 1229–1234.

[26] N. Johnson and D. Hogg, “Learning the distribution of object trajectories for event recognition,” Image and vision computing, vol. 14, no. 8, pp. 609–615, 1996.

[27] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato, “Future person localization in first-person videos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7593–7602.

[28] R. Chandra, U. Bhattacharya, C. Roncal, A. Bera, and D. Manocha, “Robustst: End-to-end trajectory prediction for heterogeneous road-agents in dense traffic with noisy sensor inputs,” in ACM Computer Science in Cars Symposium, 2019, pp. 1–9.

[29] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning: Springer series in statistics New York, 2001, vol. 1, no. 10.

[30] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.

[31] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in Advances in neural information processing systems, 2014, pp. 3104–3112.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., “Imagenet large scale visual recognition challenge,” International journal of computer vision, vol. 115, no. 3, pp. 211–252, 2015.

[33] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “Siamrpn++: Evolution of siamese visual tracking with very deep networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4292–4291.

[34] L. Zheng, M. Tang, H. Lu et al., “Learning features with differentiable closed-form solver for tracking,” arXiv preprint arXiv:1906.10414, 2019.

[35] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, “Atom: Accurate tracking by overlap maximization,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4660–4669.

[36] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, “Learning spatial-temporal regularized correlation filters for visual tracking,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4904–4913.

[37] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in European conference on computer vision. Springer, 2016, pp. 472–488.

[38] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4293–4302.

[39] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, “Eco: Efficient convolution operators for tracking,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6638–6646.
Yuan Liu received the BS degree in Optical Engineering from Nanjing University of Science and Technology, China, in 2012, and currently working towards Ph.D. degree in Nanjing University of Science and Technology. He was a visiting student with the National University of Singapore, from 2019 to 2020. His current research interests include visual object tracking and deep learning.

Ruoteng Li received the B.S. degree from National University of Singapore, Singapore, in 2013 and received his Ph.D degree from electrical and computer engineering, National University of Singapore in 2020. His research direction includes deep learning based image processing (deraining, dehazing, denoising, etc.), optical flow, stereo estimation, depth estimation and their related fields.

Robby T. Tan received the PhD degree in computer science from the University of Tokyo. He is an associate professor at both Yale-NUS College and ECE (Electrical and Computing Engineering), National University of Singapore. Previously, he was an assistant professor at Utrecht University. His research interests include computer vision and deep learning. He is a member of the IEEE.

Yu Cheng received the BS degree in Electric and Electronic Engineering from Nanyang Technological University, Singapore, in 2018, and currently working towards Ph.D. degree in National University of Singapore. His current research interests include deep-learning based human pose tracking, 2D/3D human pose estimation, face recognition and related applications.

Xiubao Sui received the PhD degree in Optical Engineering from Nanjing University of Science and Technology, China. He is currently a professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, China. His current research interests include deep-learning based image processing, image denoising, super-resolution reconstruction and related applications. He is a member of the IEEE.