

SiamRCR: Reciprocal Classification and Regression for Visual Object Tracking

Jinlong Peng^{1*}, Zhengkai Jiang^{1*}, Yueyang Gu^{1*}, Yang Wu^{2†},
Yabiao Wang¹, Ying Tai¹, Chengjie Wang¹ and Weiyao Lin³

¹Tencent Youtu Lab

²Kyoto University

³Shanghai Jiao Tong University

{jeromepeng, zhengkaijiang, yueyanggu, caseywang, yingtai, jasoncjwang}@tencent.com,
wu.yang.8c@kyoto-u.ac.jp, wylin@sjtu.edu.cn

Abstract

Recently, most siamese network based trackers locate targets via object classification and bounding-box regression. Generally, they select the bounding-box with maximum classification confidence as the final prediction. This strategy may miss the right result due to the accuracy misalignment between classification and regression. In this paper, we propose a novel siamese tracking algorithm called SiamRCR, addressing this problem with a simple, light and effective solution. It builds reciprocal links between classification and regression branches, which can dynamically re-weight their losses for each positive sample. In addition, we add a localization branch to predict the localization accuracy, so that it can work as the replacement of the regression assistance link during inference. This branch makes the training and inference more consistent. Extensive experimental results demonstrate the effectiveness of SiamRCR and its superiority over the state-of-the-art competitors on GOT-10k, LaSOT, TrackingNet, OTB-2015, VOT-2018 and VOT-2019. Moreover, our SiamRCR runs at 65 FPS, far above the real-time requirement.

1 Introduction

As one of the fundamental research topics in computer vision, visual object tracking (VOT) plays an important role in many applications such as human-computer interaction, visual surveillance, medical image processing, and so on. It aims to locate objects in subsequent sequences according to a given ground-truth for each object target in a chosen video frame where the target appears. There is no prior knowledge about object class, which is the most significant characteristic of object tracking. Although researchers have paid much attention to object tracking, it is still a challenging task when any of the following factors exists significantly: occlusion, deformation, and scale variation.

Recently, siamese network based tracking has attracted increasing interest due to its balance between accuracy and ef-

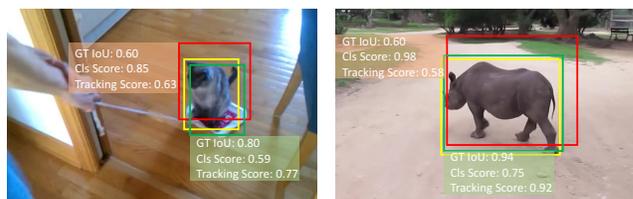


Figure 1: Case study on the accuracy misalignment problem between classification and regression of siamese network based tracking models and our solution. The yellow bounding boxes denote the ground-truths, while the red and green bounding boxes are the winners ranked by the classification score and the proposed tracking score, respectively. Clearly, the tracking scores generated by SiamRCR are much more consistent with the localization/regression accuracy values (IoU), leading to better tracking performance.

iciency [Bertinetto *et al.*, 2016; Li *et al.*, 2018; Zhang *et al.*, 2020]. A siamese network consists of two branches sharing the same parameters for feature extracting. Exemplar image (ground-truth in the first frame) and search image (ROI of a frame to be tracked in) are inputs to the siamese network. After feature extraction and cross-correlation, it breaks into two branches: a classification branch outputs a confidence map for position estimation and a regression branch predicts the target bounding box information corresponding to each position of the confidence map. Such a network structure allows a straight-forward inference method: finding the maximum value on the 2D confidence map (from the classification branch) and then using its position to get the corresponding regressed bounding box information (from the regression branch). However, such a siamese structure generally has classification and regression optimized independently and all existing models have failed to make them properly synchronized. This results in the **accuracy misalignment between classification and regression**. As shown in Figure 1, the predicted box with high classification confidence may not have high regression accuracy in terms of IoU (Interaction over Union) score. Due to the misalignment, the bounding-box which locates the target more accurately than others might be discarded, leading to an inferior tracking performance. Although some recent siamese network [Danelljan *et al.*, 2019; Xu *et al.*, 2020] have tried to predict the localization/regression accuracy, the misalignment over there is still

*Equal contribution.

†Corresponding author: Yang Wu (wuyang0321@gmail.com).

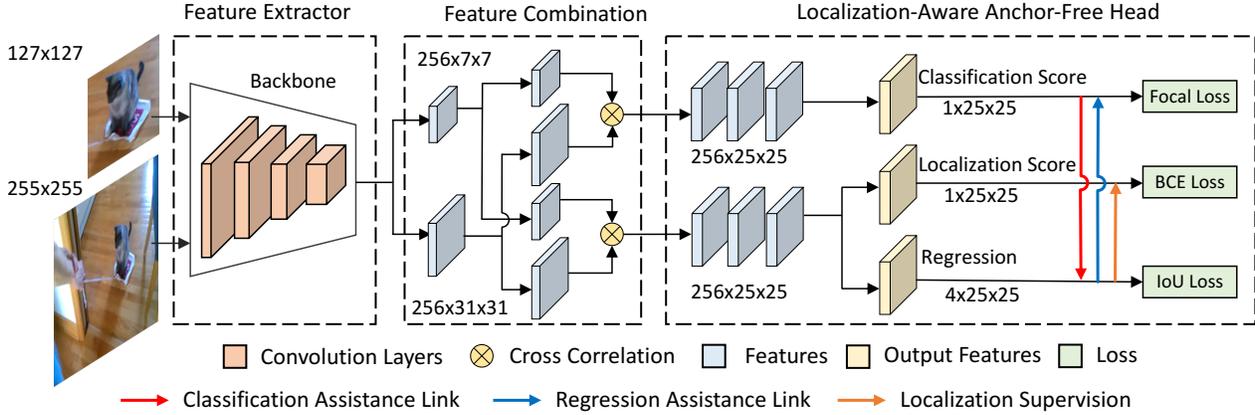


Figure 2: Our proposed siamese framework on Reciprocal Classification and Regression (SiamRCR). It consists of a feature extractor, a feature combination module, and a three-branch siamese head structure with the novel reciprocal links over the individual losses. Note that the three links between the three branches are only designed for loss calculation during training and do not exist during inference.

severe since the independent optimization issue of classification and regression remains unsolved.

In this paper, we propose a novel solution to alleviate the misalignment, which builds a reciprocal relationship between classification and regression, so that they can be optimized in a synchronized way for generating accuracy consistent outputs. Since the reciprocal relationship is the key for its success, we name our model Siamese Network based Reciprocal Classification and Regression with **SiamRCR** as its abbreviation. The overall framework of SiamRCR is shown in Figure 2. Besides the commonly used classification branch and regression branch, we add two links (the classification assistance link and the regression assistance link) to build the reciprocal relationship between them during model training. Classification assists regression by weighting the regression loss with the classification confidence, so that regression can focus more on high confident positions for more precise location. Regression assists classification by weighting the classification loss with the localization score derived from the regressed bounding box and the ground-truth box, forcing classification score to be more consistent with regression accuracy. Since there is no such localization score during testing/inference (ground-truth bounding box is unknown), a localization branch is added to predict such a localization score at each position, so that the prediction can be used as localization score’s approximation to be consistent with the training model. Therefore, the multiplication of the classification confidence and the localization prediction confidence generates a new tracking score/confidence map for regression during testing, which ensures the consistency with the training process.

Besides the key idea of reciprocal classification and regression, two other designs also contribute to the effectiveness and superiority of our model. One is that we choose to build on the anchor-free tracking mechanism so that the whole model can be one-stage, clean, efficient with fewer hyper-parameters. The other is that our model predicts center offset and width/height of the target, which is more straightforward and efficient than other VOT methods.

The main contributions of this work are listed as follows:

1. We propose a novel tracking model that solves the long-standing unsolved classification and regression misalignment problem, with new simple, intuitive and efficient designs.
2. It presents a new way on how to link losses of multiple branches and make the training and inference process more consistent, which may provide inspirations to other tasks.
3. Our SiamRCR achieves state-of-the-art performance on six public benchmarks, including GOT-10k, TrackingNet, LaSOT, OTB-2015, VOT-2018 and VOT-2019. The framework is built on an anchor-free mechanism with a more direct center offset and width/height prediction, running at 65 FPS.

2 Related Works

2.1 Siamese Network based Framework

Comparing with traditional correlation filter tracking methods, recent siamese network based methods have achieved superior performance since the pioneering work SiamFC was proposed [Bertinetto *et al.*, 2016]. More recent studies [Li *et al.*, 2018; Li *et al.*, 2019a] try to introduce object detection progresses into object tracking for more accurate location prediction. Though these works have explored several important aspects, the accuracy misalignment problem between classification and regression has been overlooked. Ocean [Zhang *et al.*, 2020] partially concerns a similar issue and presents a feature alignment module to alleviate it by utilizing the prediction of regression branch to refine the classification branch. However, this cannot eliminate the misalignment problem as the alignment is unidirectional. Differently, our SiamRCR focuses on the misalignment problem and proposes a simple, intuitive and more thorough solution with bidirectional and reciprocal links and a novel complementary branch for making training and inference consistent.

2.2 Anchor-Free Tracking Mechanism

Anchor-free methods have recently attracted widespread attention in the object detection field [Law and Deng, 2018; Duan *et al.*, 2019; Tian *et al.*, 2019; Zhou *et al.*, 2019] due to their simplicity and efficiency. Naturally, the anchor-free

mechanism has also been introduced to the tracking field [Xu *et al.*, 2020; Chen *et al.*, 2020; Zhang *et al.*, 2020]. Multiple object tracking (MOT) is a related area of VOT [Peng *et al.*, 2020a; Peng *et al.*, 2020c]. In MOT area, based on CenterNet [Zhou *et al.*, 2019], CenterTrack [Zhou *et al.*, 2020] obtains high performance by predicting the center point, width/height and center offset of each object. To our best knowledge, SiamRCR is the first VOT method predicting center offset and width/height of the target, which is more straightforward and efficient than ever.

2.3 Dynamic Sample Re-weighting

Existing trackers [Li *et al.*, 2018; Li *et al.*, 2019a; Xu *et al.*, 2020; Peng *et al.*, 2020b] directly use some heuristic rules, *e.g.*, the Focal Loss [Lin *et al.*, 2017] to define the labels of samples and their weights. PrDiMP [Danelljan *et al.*, 2020] models the uncertainty of the labels. Such predefined static weights lead to the accuracy misalignment problem between classification and regression, which harms the final tracking accuracy. However, in our SiamRCR, the sample weights for each loss become dynamic as they are conditioned on the other branch’s outputs which keep changing during the interaction. Such dynamic sample re-weighting mechanism is novel and also critical to the effectiveness of our model.

2.4 Localization Prediction Strategy

In object detection area, IoU-Net [Jiang *et al.*, 2018] predicts the IoU between each detected box and the matched ground-truth to guide the box regression, which is class-specific thus not directly suitable for VOT. ATOM [Danelljan *et al.*, 2019] trains a target-specific IoU prediction network offline and SiamFC++ [Xu *et al.*, 2020] estimates the bounding box quality based on centerness [Tian *et al.*, 2019]. However, both the purpose and implementation of the localization branch in our SiamRCR are different. Our localization branch is a natural auxiliary of the reciprocal classification and regression structure which itself is a better solution than existing works, while the IoU network in other works is the main. Moreover, our localization branch is simple and lightweight, which ensures the effectiveness and efficiency of the algorithm simultaneously.

3 Proposed Method

3.1 Overview

The proposed siamese tracking framework is shown in Figure 2. Different from previous anchor-based [Li *et al.*, 2018; Li *et al.*, 2019a] methods which rely on pre-defined anchor sizes and scales, our method is anchor-free. It operates as follows. First, the target template and the current frame are both fed into the shared feature extractor (using the backbone of [He *et al.*, 2016]) to generate their corresponding features. Then, such features are combined through depth-wise cross-correlation operation to create correlated feature maps, which are further fed into the corresponding classification and regression branches of the anchor-free tracking head. The built-in reciprocal links dynamically re-weight the samples for computing each loss of the two branches. A new localization branch grows from the regression branch for predicting

the localization accuracy. Its output can serve as the approximation of the localization score during inference to generate a more accurate tracking score together with the classification confidence. The key components are in detail as below.

3.2 Anchor-Free Tracking with Box Regression

For the i -th input pair from the training set, we have $F_i \in \mathcal{R}^{C \times H \times W}$ denotes the feature map of the classification branch and s be the total stride. The ground-truth bounding box for the current frame is defined as $B_{x,y}^* = (x_0^*, y_0^*, x_1^*, y_1^*)$, *i.e.*, coordinates of the bounding box. For each location (x, y) on the feature map F_i , we can map it back onto the input frame to get the corresponding image coordinates $(\lfloor \frac{x}{s} \rfloor + xs, \lfloor \frac{y}{s} \rfloor + ys)$. Different from anchor-based trackers, which consider the location on the input frame as the center of anchor boxes and regress the target bounding boxes w.r.t. the anchor boxes, we directly regress the target boxes’ width and height values and the center offsets at the location. In this way, our tracker views locations as training samples instead of anchor boxes, which follows the paradigm of the FCNs [Long *et al.*, 2015] for semantic segmentation.

Specially, the sample at location (x, y) is considered to be positive if it falls into a radius r at the ground-truth box center, and the radius is a hyper-parameter for the proposed method. Otherwise, it is a negative sample (background). Besides the label (denoted by $c_{x,y}^*$) for foreground-background classification, we also have a 4D real vector $t_{x,y}^* = (w^*, h^*, \Delta x^*, \Delta y^*)$ indicating the regression target for the localization. Here, w^* and h^* are the width and height of target ground-truth bounding box, while Δx^* and Δy^* are the center offsets between the current location and the ground-truth box. Formally, if location (x, y) is associated to the ground-truth box $B_{x,y}^*$, which has width w^* and height h^* , then we have

$$\begin{aligned} w^* &= x_1^* - x_0^*, & h^* &= y_1^* - y_0^*, \\ \Delta x^* &= (x_0^* + x_1^*)/2 - x, & \Delta y^* &= (y_0^* + y_1^*)/2 - y. \end{aligned} \quad (1)$$

Corresponding to the training target, SiamRCR predicts a classification confidence score $p_{x,y}^{cls}$, a regressed 4D vector $t_{x,y} = (w, h, \Delta x, \Delta y)$ for the bounding box, and a localization confidence score $p_{x,y}^{loc}$ denoting the predicted localization accuracy. It is worth noting that SiamRCR has $5 \times$ fewer network parameters than the popular anchor-based tracker SiamRPN [Li *et al.*, 2018] with 5 anchor boxes per location.

3.3 Reciprocal Classification and Regression

In existing siamese network tracking models, classification and regression branches operate in parallel and get optimized independently with their own losses, which aggravates the accuracy misalignment of their results. In fact, when a regressed bounding box has low accuracy, the corresponding classification score should not be high, because if that position becomes the winner of classification confidence the bad localization will lead to bad tracking performance. And when a bounding box has a low classification score, there is no meaning for the regression to try hard to get a high localization accuracy for it will not be the winner anyway. Therefore, these two branches need to talk to each other for aligning the accu-

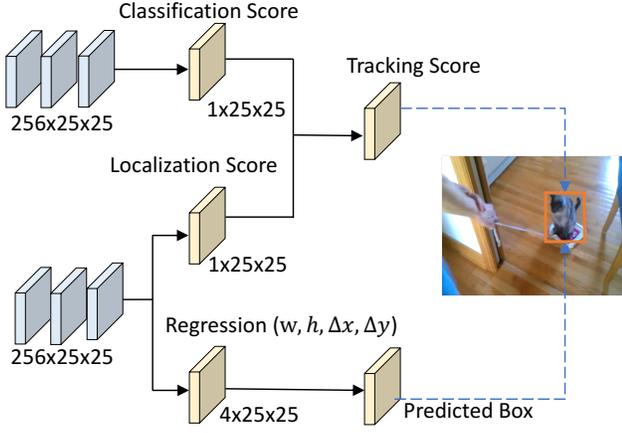


Figure 3: The head of SiamRCR during inference. The classification score and localization score are multiplied to generate the final tracking score for ranking the predicted bounding boxes.

racy of their results. In this paper, we propose a novel strategy called reciprocal classification and regression to make these two branches assist each other. It is implemented by building two links, including regression assistance link and classification assistance link.

Regression Assistance Link

To eliminate the chance that low localization accuracy bounding boxes still get high classification scores, a simple yet effective solution is to use the localization accuracy to weight the classification loss. Such an assistance link from regression can be regarded as a kind of dynamic sample re-weighting as the localization accuracy keeps changing during the model optimization. The dynamically re-weighted classification loss can be formulated as:

$$L_{cls} = \frac{1}{N_{pos}} \sum_{x,y} L_{Focal}(p_{x,y}^{cls}, c_{x,y}^*) * IoU(B_{x,y}, B_{x,y}^*), \quad (2)$$

where L_{Focal} and IoU denote the focal loss [Lin *et al.*, 2017] and the IoU score, respectively, N_{pos} is the number of positive samples, and $B = (x_0, y_0, x_1, y_1)$ is the predicted bounding box at location (x, y) with predicted width/height (w, h) and center offsets $(\Delta x, \Delta y)$:

$$\begin{aligned} x_0 &= x + \Delta x - w/2, & y_0 &= y + \Delta y - h/2, \\ x_1 &= x + \Delta x + w/2, & y_1 &= y + \Delta y + h/2. \end{aligned} \quad (3)$$

Classification Assistance Link

To avoid low confidence positions getting highly accurate bounding boxes, the regression branch should be aware of the classification confidence. To this end, $p_{x,y}^{cls}$ is utilized to dynamically re-weight the regression loss as:

$$L_{reg} = \frac{1}{N_{pos}} \sum_{x,y} \mathbb{I}_{\{c_{x,y}^*=1\}} L_{IoU}(t_{x,y}, t_{x,y}^*) * p_{x,y}^{cls}, \quad (4)$$

where L_{IoU} is the IoU loss as in UnitBox [Yu *et al.*, 2016]; $\mathbb{I}_{\{c_{x,y}^*=1\}}$ is an indicator function, which equals to 1 if $c_{x,y}^* = 1$ and 0 otherwise.

Localization Score Branch

The regression assistance link makes the classification branch aware of the regression accuracy during training, thanks to the ground-truth bounding box $B_{x,y}^*$ for computing the localization score. However, in the inference stage there is no such ground-truth. Directly using the classification confidence map p^{cls} to select the winner bounding box may still lead to certain accuracy misalignment, as the localization score was hands-on during the classification branch’s training. The hands-on inductive training makes p^{cls} collaborative with the localization score but not necessarily consistent with it. Therefore, we let the regression branch grow a new branch called localization branch to be trained for predicting the localization score given the feature maps for regression, under the following loss function.

$$L_{loc} = \frac{1}{N_{pos}} \sum_{x,y} \mathbb{I}_{\{c_{x,y}^*=1\}} L_{BCE}(p_{x,y}^{loc}, IoU(B_{x,y}, B_{x,y}^*)), \quad (5)$$

where L_{BCE} is the Binary Cross Entropy (BCE) loss.

As shown in Figure 3, during inference, the final tracking score (used for ranking the predicted bounding boxes) is computed by multiplying $p_{x,y}^{cls}$ with $p_{x,y}^{loc}$, making the inference localization-aware. Thus, the localization branch can further reduce the low-quality boxes and improve the overall tracking accuracy.

The Overall Training Objective

With the above losses for SiamRCR’s three branches, we can define its final training loss function as:

$$L = L_{cls} + \lambda_1 * L_{reg} + \lambda_2 * L_{loc}. \quad (6)$$

where λ_1 and λ_2 are the hyper-parameters for balancing these losses. In our experiments, they are all set to 1.

4 Experiments

4.1 Implementation Details

Training Phase. We utilize ResNet-50 [He *et al.*, 2016] as the backbone of our SiamRCR. We remove the last conv-block for higher resolution feature map and utilize dilated convolution for higher receptive field [Li *et al.*, 2019a]. The backbone is initialized with the parameters pre-trained on ImageNet [Russakovsky *et al.*, 2015]. The whole network is optimized by Stochastic Gradient Descent (SGD) with momentum 0.9 on the datasets of GOT-10k [Huang *et al.*, 2019], TrackingNet [Müller *et al.*, 2018], COCO [Lin *et al.*, 2014], LaSOT [Fan *et al.*, 2019], ImageNet VID [Russakovsky *et al.*, 2015] and ImageNet DET [Russakovsky *et al.*, 2015]. We totally train the network for 20 epochs. The batch size is 128. The learning rate is from 0.000001 to 0.1 in the first 5 epochs for warm-up and from 0.1 to 0.0001 with cosine schedule in the last 15 epochs. We freeze the backbone in the first 10 epochs and fine-tune it in the other 10 epochs with a reduced learning rate (multiplying 0.1). The size of exemplar image and search image are 127*127 and 255*255, respectively. Our algorithm is implemented by Python 3.6 and PyTorch 1.1.0. The experiments are conducted on a server with Intel(R) Xeon(R) CPU E5-2680 v4 2.40GHz, and a NVIDIA Tesla P40 24GB GPU with CUDA 10.1.

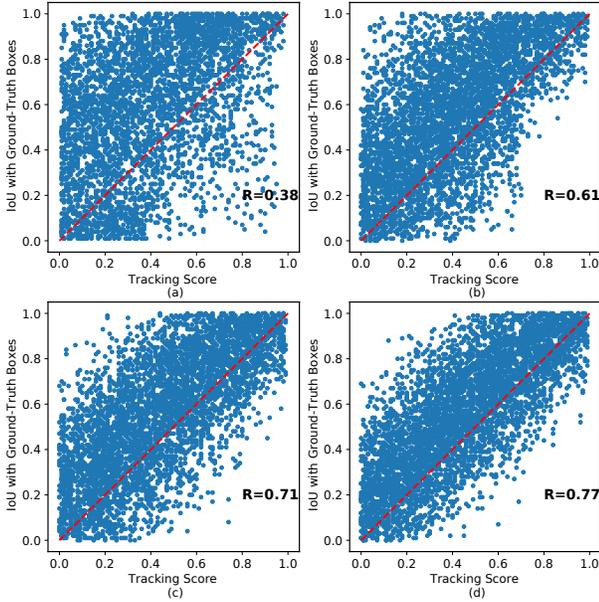


Figure 4: The correlation between IoU scores and the tracking score, together with the Pearson correlation coefficient R . (a) Baseline model, the tracking score is the classification score. (b) Using centerness proposed in FCOS [Tian *et al.*, 2019] as the tracking score. (c) Baseline + localization branch. (d) SiamRCR.

Testing Phase. We utilize the same offline testing strategy as [Xu *et al.*, 2020]. The ground-truth after augmentation in the first frame is used as the exemplar image and we keep it unchanged during the whole testing phase. A cosine-window [Bertinetto *et al.*, 2016] is multiplied on the confidence map. We adopt a linear interpolation updating strategy on scale prediction to make the final box change smoothly over time. We evaluate SiamRCR on six public benchmarks following their corresponding protocols: GOT-10k [Huang *et al.*, 2019], TrackingNet [Müller *et al.*, 2018], LaSOT [Fan *et al.*, 2019], OTB-2015 [Wu *et al.*, 2015], VOT-2018 [Kristan *et al.*, 2018] and VOT-2019 [Kristan *et al.*, 2019].

4.2 Ablation Study

Component. The ablation study results on the key components of SiamRCR are presented in Table 1. The baseline (I) without localization branch and reciprocal links obtains an AO (Average Overlap) of 0.594. With localization branch, SiamRCR can predict the localization score of the regressed bounding box, making the final tracking score more consistent with the real IoU than the classification score. Multiplying the localization score alone (II) improves the performance by 3.54% compared with baseline, showing the significance of the accuracy misalignment between classification and regression. Building reciprocal assistance links itself (III) can also gain a relative improvement of 2.86% over the baseline, proving that the misalignment can be alleviated between classification and regression. When these two components are both adopted, the relative performance is more remarkable: 5.05%, which is nearly equal to the direct sum of both performance gains. It confirms that the localization branch is con-

	Localization Branch	Reciprocal Links	AO \uparrow
I			0.594
II	✓		0.615
III		✓	0.611
IV	✓	✓	0.624

Table 1: Ablation study on GOT-10k test set.

r	R	AO \uparrow	SR@0.5 \uparrow	SR@0.75 \uparrow
1	8	0.593	0.723	0.458
2	16	0.624	0.752	0.460
3	24	0.619	0.747	0.459
4	32	0.612	0.743	0.446
5	40	0.611	0.740	0.474

Table 2: Comparative experiment in terms of r on GOT-10k test set.

sistent with the reciprocal links, serving well as the replacement of the regression assistance link for inference. To better demonstrate how well our SiamRCR alleviates the accuracy misalignment problem, we illustrate the correlation between the IoU of regressed bounding box (w.r.t. the matched ground-truth) and the tracking score in Figure 4. As shown in Figure 4(a), the Pearson correlation coefficients between IoU and tracking score is only 0.38, showing that the classification score is indeed not consistent with the real localization accuracy. Figure 4(c) and 4(d) show that both the localization branch and the reciprocal links are effective and necessary, and they can well collaborate with each other.

Predicted IoU vs. Centerness. Centerness is pre-defined label which indicates the distance between candidates and target center. Some object detection [Tian *et al.*, 2019] or object tracking [Xu *et al.*, 2020] algorithms utilize centerness to assist localization. In our SiamRCR, we discard this kind of fixed prior and utilize predicted IoU as dynamic supervised localization information. Thus, our localization branch can estimate the localization confidence more accurately. As shown in Figure 4 (b) and (c), our localization prediction mechanism alleviates the misalignment between classification and regression, which is better than centerness.

Radius. Radius r is a significant hyper-parameter in our proposed anchor-free framework. It decides the division of positive samples and negative samples during training. We conduct comparative experiment in terms of r . The results are shown in Table 2. R is the corresponding radius of r in the original input video frame, which is 8 times r . When $r = 1$, the performance on GOT-10k is poor since the number of positive samples is too small. When $r = 2$, our SiamRCR achieves the best performance. When $r = 4$ or $r = 5$, the positive samples are redundant since some candidates far from target center are divided into positive samples. Therefore, the performance drops compared with $r = 2$ or $r = 3$.

4.3 Comparison with the State-of-the-Art

We compare our SiamRCR with 18 state-of-the-art trackers. The datasets and experimental settings are detailed as below. Due to space limitations, the experiments on OTB-2015 and

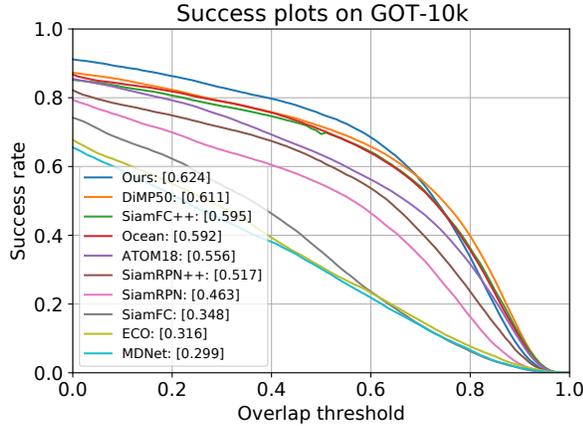


Figure 5: Comparison of tracking results on GOT-10k benchmark.

	Succ.↑	Prec.↑	N-Prec.↑
SiamFC [Bertinetto <i>et al.</i> , 2016]	0.559	0.518	0.652
ECO [Danelljan <i>et al.</i> , 2017]	0.554	0.492	0.618
UPDT [Zhang <i>et al.</i> , 2019]	0.611	0.557	0.702
ATOM [Danelljan <i>et al.</i> , 2019]	0.703	0.648	0.771
SiamRPN++ [Li <i>et al.</i> , 2019a]	0.733	0.694	0.800
DiMP50 [Bhat <i>et al.</i> , 2019]	0.740	0.687	0.801
KYS [Bhat <i>et al.</i> , 2020]	0.740	0.688	0.800
SiamAttn [Yu <i>et al.</i> , 2020]	0.752	0.715	0.817
SiamFC++ [Xu <i>et al.</i> , 2020]	0.754	0.705	0.800
SiamRCR (ours)	0.764	0.716	0.818

 Table 3: Comparison of tracking results on TrackingNet benchmark. **Red** and **blue** fonts indicate the best and second results respectively.

VOT-2018 are presented in the supplementary.

GOT-10k. The evaluation follows the protocols in [Huang *et al.*, 2019]. For a fair comparison, we train SiamRCR only on the *train* subset which consists of about 10,000 sequences and test it on the *test* subset of 180 sequences. As shown in Figure 5, our SiamRCR achieves 0.624 of AO, which is the best among evaluated trackers (including the online updating tracker DiMP). The slightly inferior performance at large overlap threshold might due to SiamRCR’s strategy of predicting the center offsets and width/height, rather than predicting the bounding box coordinate offsets (e.g. SiamFC++), as larger value ranges can lead to less preciseness. However, our strategy better solves the misalignment problem.

TrackingNet. The *test* subset of it contains 511 sequences and 70 object classes. We also train our model only on TrackingNet *train* subset. There are three metrics in TrackingNet: Success (Succ.), Precision (Prec.) and Normalized Precision (N-Prec.). We report the results in Table 3. SiamRCR surpasses other state-of-the-art trackers on all three evaluation metrics. In particular, SiamRCR obtains 0.764 of Succ., 0.716 of Prec. and 0.818 of N-Prec., which further demonstrates the superior tracking performance of our SiamRCR.

LaSOT. LaSOT is a large-scale long-term tracking benchmark. It contains 1,400 sequences and more than 3.5 million frames. We train our model only on LaSOT *train* sub-

	Succ.↑	Prec.↑
SiamFC [Bertinetto <i>et al.</i> , 2016]	0.339	0.336
MDNet [Nam and Han, 2016]	0.373	0.397
GradNet [Li <i>et al.</i> , 2019b]	0.351	0.365
SiamRPN++ [Li <i>et al.</i> , 2019a]	0.491	0.496
ATOM [Danelljan <i>et al.</i> , 2019]	0.505	0.514
DiMP50 [Bhat <i>et al.</i> , 2019]	0.564	0.568
ROAM++ [Yang <i>et al.</i> , 2020]	0.447	0.445
SiamBAN [Chen <i>et al.</i> , 2020]	0.514	0.518
Ocean [Zhang <i>et al.</i> , 2020]	0.526	0.526
SiamFC++ [Xu <i>et al.</i> , 2020]	0.544	0.547
SiamRCR (ours)	0.575	0.599

Table 4: Comparison of tracking results on LaSOT benchmark.

	EAO↑	Accuracy↑	Robustness↓
SPM [Wang <i>et al.</i> , 2019a]	0.275	0.577	0.507
SiamRPN++ [Li <i>et al.</i> , 2019a]	0.285	0.599	0.482
SiamMask [Wang <i>et al.</i> , 2019b]	0.287	0.594	0.461
SiamBAN [Chen <i>et al.</i> , 2020]	0.327	0.602	0.396
Ocean [Zhang <i>et al.</i> , 2020]	0.327	0.590	0.376
SiamRCR (ours)	0.336	0.602	0.386

Table 5: Comparison of tracking results on VOT-2019 benchmark.

set and conduct evaluation following the protocol II in [Fan *et al.*, 2019]. As shown in Table 4, our SiamRCR achieves 0.575 of Succ. and 0.599 of Prec., and outperforms recent SOTA tracker Ocean by 8.5% and 13.9% in terms of both Success and Precision score respectively. It also achieves better performance compared with other localization-aware trackers (ATOM and SiamFC++), proving that our reciprocal links with localization branch is better.

VOT-2019. With challenging factors such as occlusion, fast motion and illumination changing in 60 test sequences, VOT-2019 provides a comprehensive evaluation platform for VOT. Commonly used metrics for it are Expected Average Overlap (EAO), Accuracy and Robustness. EAO takes both Accuracy and Robustness into account to verify the overall tracking performance. We report experimental results on VOT-2019 in Table 5. Our SiamRCR achieves the best EAO score, the best Accuracy score and the second best Robustness score. Ocean performs slightly better in Robustness with the multi-feature combination strategy. As our SiamRCR only uses single conv-feature for estimation, it is faster than Ocean. Moreover, it demonstrates superior effectiveness and efficiency.

5 Conclusion

In this paper, we have proposed a novel anchor-free object tracking framework which is efficient and effective. It addresses the long-term standing accuracy misalignment problem of Siamese network based models. Elaborate ablation studies have shown the effectiveness of the whole proposed model and its key components. Without bells and whistles, the proposed method achieves state-of-the-art performance on six tracking benchmarks, with a running speed of 65 FPS.

References

- [Bertinetto *et al.*, 2016] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.
- [Bhat *et al.*, 2019] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019.
- [Bhat *et al.*, 2020] Goutam Bhat, Martin Danelljan, Luc Gool, Van, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020.
- [Chen *et al.*, 2020] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, 2020.
- [Danelljan *et al.*, 2017] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ECO: efficient convolution operators for tracking. In *CVPR*, 2017.
- [Danelljan *et al.*, 2019] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: accurate tracking by overlap maximization. In *CVPR*, 2019.
- [Danelljan *et al.*, 2020] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *CVPR*, 2020.
- [Duan *et al.*, 2019] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *CVPR*, 2019.
- [Fan *et al.*, 2019] Heng Fan, Liting Lin, Fan Yang, et al. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [Huang *et al.*, 2019] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *TPAMI*, 2019.
- [Jiang *et al.*, 2018] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.
- [Kristan *et al.*, 2018] Matej Kristan, Ales Leonardis, Jiri Matas, et al. The sixth visual object tracking vot2018 challenge results. In *ECCV*, 2018.
- [Kristan *et al.*, 2019] Matej Kristan, Jiri Matas, Ales Leonardis, et al. The seventh visual object tracking vot2019 challenge results. In *ICCVW*, 2019.
- [Law and Deng, 2018] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.
- [Li *et al.*, 2018] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.
- [Li *et al.*, 2019a] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019.
- [Li *et al.*, 2019b] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Gradnet: Gradient-guided network for visual object tracking. In *ICCV*, 2019.
- [Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *CVPR*, 2017.
- [Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [Müller *et al.*, 2018] Matthias Müller, Adel Bibi, Silvio Giancola, et al. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.
- [Nam and Han, 2016] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.
- [Peng *et al.*, 2020a] Jinlong Peng, Yueyang Gu, Yabiao Wang, Chengjie Wang, Jilin Li, and Feiyue Huang. Dense scene multiple object tracking with box-plane matching. In *ACM MM*, 2020.
- [Peng *et al.*, 2020b] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *ECCV*, 2020.
- [Peng *et al.*, 2020c] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *PR*, 2020.
- [Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Su Hao, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.
- [Tian *et al.*, 2019] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.
- [Wang *et al.*, 2019a] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, 2019.
- [Wang *et al.*, 2019b] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.
- [Wu *et al.*, 2015] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 2015.
- [Xu *et al.*, 2020] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, 2020.
- [Yang *et al.*, 2020] Tianyu Yang, Pengfei Xu, Runbo Hu, Hua Chai, and Antoni Chan. Roam: Recurrently optimizing tracking model. In *CVPR*, 2020.
- [Yu *et al.*, 2016] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACM MM*, 2016.
- [Yu *et al.*, 2020] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R. Scott. Deformable siamese attention networks for visual object tracking. In *CVPR*, 2020.
- [Zhang *et al.*, 2019] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, 2019.
- [Zhang *et al.*, 2020] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu1. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.
- [Zhou *et al.*, 2019] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv:1904.07850*, 2019.
- [Zhou *et al.*, 2020] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020.