Ranking-Based Siamese Visual Tracking

Feng Tang[1,2] Qiang Ling[1*]
[1] Department of Automation, University of Science and Technology of China, China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, China
tang0420@mail.ustc.edu.cn, qling@ustc.edu.cn

Abstract

Current Siamese-based trackers mainly formulate the visual tracking into two independent subtasks, including classification and localization. They learn the classification subnetwork by processing each sample separately and neglect the relationship among positive and negative samples. Moreover, such tracking paradigm takes only the classification confidence of proposals for the final prediction, which may yield the misalignment between classification and localization. To resolve these issues, this paper proposes a ranking-based optimization algorithm to explore the relationship among different proposals. To this end, we introduce two ranking losses, including the classification one and the IoU-guided one, as optimization constraints. The classification ranking loss can ensure that positive samples rank higher than hard negative ones, i.e., distractors, so that the trackers can select the foreground samples successfully without being fooled by the distractors. The IoU-guided ranking loss aims to align classification confidence scores with the Intersection over Union(IoU) of the corresponding localization prediction for positive samples, enabling the well-localized prediction to be represented by high classification confidence. Specifically, the proposed two ranking losses are compatible with most Siamese trackers and incur no additional computation for inference. Extensive experiments on seven tracking benchmarks, including OTB100, UAV123, TC128, VOT2016, NFS30, GOT-10k and LaSOT, demonstrate the effectiveness of the proposed ranking-based optimization algorithm. The code and raw results are available at https://github.com/sansanfree/RBO.

1. Introduction

Visual object tracking aims to estimate the location information of an arbitrary target in each frame of a video sequence. In most situations, only the initial target information is provided for trackers, then trackers are required to model the target appearance in the next frames. Since target-specific information is only available at test-time, the target model cannot be obtained via offline network training.

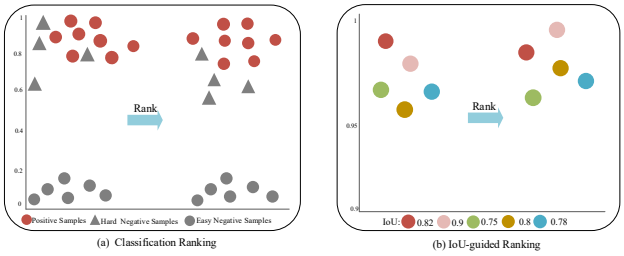

Figure 1. Illustration of the proposed two ranking losses. The classification loss enables positive samples to rank higher than hard negative samples, aiming to suppress the classification confidence of distractors. The IoU ranking loss aims to align classification with localization, i.e., that the samples with larger IoUs are expected to gain higher classification confidence scores.

Recently, many researchers exploit how to take the power of deep learning technology to solve the tracking task. Siamese network is one of the most popular deep learning paradigm. As pioneer work, SiamFC [1] formulates visual tracking into a deep target matching problem. To be specific, SiamFC consists of two branches, i.e., target template and search region. Its former branch is used to model the target as a fixed exemplar and the later branch processes the possible regions. SiamFC inspires many later trackers [26,27,46,50] that are built upon the Siamese network architecture and can achieve state-of-the-art performance. Among them, SiamRPN introduces region proposal networks(RPN) consisting of a classification head for foreground-background discrimination and a regression head for anchor refinement. SiamRPN++ [26] and SiamFC++ [50] unleash the capability of deeper backbone networks, such as ResNet [21] and GoogleNet [37], to enhance feature representation. Inspired by anchor-free object detectors like FCOS [39] and CornerNet [25], many anchor-free tracker-

*Corresponding Author.

s [8, 14, 18, 19] follow the pixel-wise prediction fashion to perform the target localization.

Although Siamese based trackers have achieved promising performance, there still suffer from two limitations: (1) Siamese trackers have difficulty in distinguishing background distractors. In particular, at the training stage, the classification subnetwork is optimized by a large number of training samples, among which there exist vast uninformative samples(i.e., easy sample) that can be easily classified while a handful of distracting examples are inundated and contribute to the minor effect on network optimization. At the test-time, although the most non-target samples could be discriminated by the trackers, a background distractor could heavily mislead the tracker when it has strong positive confidence, yielding tracking failure. (2) There exists the mismatch problem between classification and localization as the two tasks are processed separately. More specifically, the classification loss drives the model to distinguish the concerned target from background regardless of the location information, while the regression branch aims at localizing the target's bounding box for all positive samples without taking into account the classification information. As a result, well-localized proposals may have relatively lower foreground confidence while the proposals with high foreground confidence may yield low localization accuracy.

To resolve the above issues, we propose a ranking-based optimization(RBO) consisting of both classification and IoU-guided ranking losses. The classification loss is used to explicitly model the relationship between the positive and hard negative samples. Actually, there are many diverse sample reweighing strategies to suppress the distractors in object detection [4, 5, 30]. However, in the context of visual tracking, hard negative samples always have the same semantic class with the tracked target, and it is difficult to discriminate the distractors in the classification embedding space. Differently, as shown in Figure 1(a), we tackle the classification as a ranking task where the foreground samples are encouraged to rank higher than background samples. Compared with the original classification loss, the proposed ranking optimization serves a loose constraint, under which hard negative samples are allowed to be classified as the foreground as long as their foreground confidence scores are lower than those of positive ones, and can well prevent the tracker from being fooled by distractors. Actually, the misalignment problem between classification and localization was studied in object detection [31, 43, 55], inspired by which we propose an IoU-guided ranking loss on the basis of RankDetNet [31] to align the foreground confidence scores of with their corresponding IoU values as shown by Figure 1(b). The modified loss is more suitable for tracking task, which enables classification confidence to be localization-sensitive to some extent.

To verify the effectiveness of the proposed ranking-based optimization, we choose anchor-based SiamRPN++ [26] and anchor-free SiamBAN [8] trackers as baselines, and craft our SiamRPN++-RBO and SiamBAN-RBO trackers, respectively. Moreover, recent works propose diverse pixel-wise correlation ways to calculate the similarity between the exemplar and search images [17, 18, 29]. Inspired by them, we modify the SiamBAN-RBO by replacing the depth-wise correlation with the pixel-wise correlation, obtaining a new tracker version called SiamPW-RBO. The main contributions of this paper are summarized as follows.

- We devise a classification ranking loss to enhance the discrimination capability by modeling the relationship between foreground samples and background ones, which prevents the tracker from being fooled by distractors.
- We propose an IoU-guided ranking loss to alleviate the mismatch problem between classification and localization. It connects two independent subtasks by aligning the classification score with associated IoU, ensuring that well-localized prediction could be represented by high classification confidence score.
- The proposed RBO can significantly improve the performance of three types of trackers on seven diverse benchmarks without sacrificing inference speed compared with baseline trackers.

2. Related Work

2.1. Siamese visual tracking

Recently, SiamFC [1] formulates the visual tracking task as a general similarity computation problem between the target template and the search region, which learns a general discriminator via large-scale offline training. Subsequent trackers have been proposed to further enhance the Siamese framework by introducing an attention mechanism [46], designing a new network architecture [42], using an enhanced loss [12], or utilizing deep reinforcement learning [44]. In these follow-ups, it is worth mentioning that SiamRPN [27] introduces the RPN module to predict bounding boxes of the targets undergoing variation in aspect ratio, instead of the brute-force discrete scale search strategy of SiamFC. Hence, SiamRPN upgrades SiamFC into an advanced framework. Based on SiamRPN, various trackers were proposed. Among them, DaSiamRPN [60] collects more diverse training data to enhance discrimination. C-RPN [16] constructs multi-stage RPNs to perform state estimation more accurately. SiamRPN++ [26] adopts a deeper ResNet-50 [21] network to enhance feature representation. Inspired by the anchor-free object detection and instance segmentation, some trackers modified the original RPN architecture into pixel-wise tracking [8, 18, 19, 50].

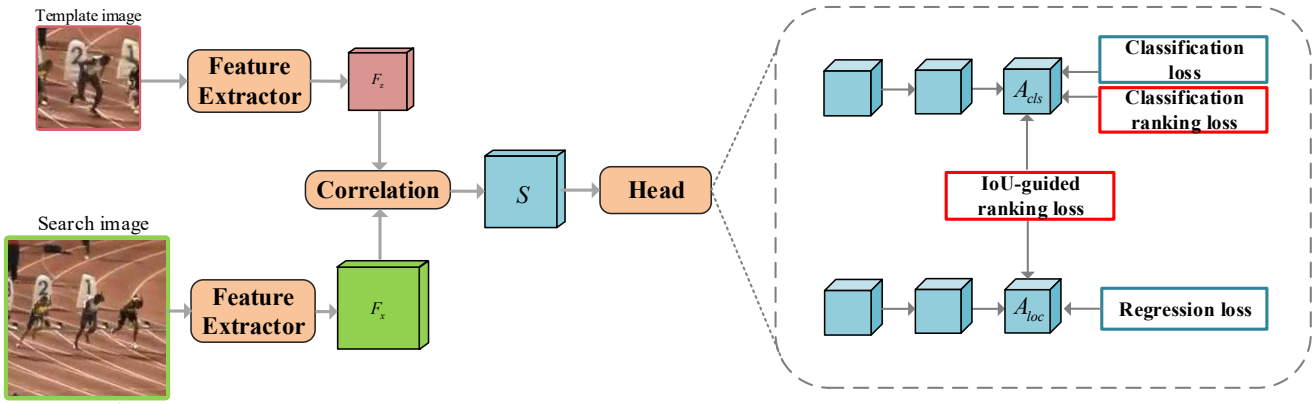

Figure 2. The pipeline of the Siamese network based tracker consists of two subtasks, including classification and localization. The proposed classification ranking loss facilitates the classification optimization while the IoU-guided ranking loss aims to align classification confidence with localization prediction.

Besides that, [17, 18, 29, 54] focus on facilitating similarity learning via modification on non-local attention [48].

Although the above Siamese trackers have achieved satisfactory performance, they can be easily misled by distractors, i.e., the robustness of the trackers can be weak. To resolve this robustness issue, many researchers introduce online deep learning techniques to enhance the generalization capability of trackers. For example, ATOM [11] and DiMP [2] construct a target-special classifier at each online tracking, and collect the historical hard negative samples to boost the classifier. UpdateNet [56] updates the target template to incorporate the temporal information. MAML [41] transfers object detectors, such as FCOS [39], to trackers via meta-learning and incrementally updates the network weights to adapt the target-special sequences. However, those trackers need to design online tracking protocols carefully to avoid false and redundant updates. Differently, this paper proposes a rank-based classification loss to suppress distractors and only involves offline training without modification on network architectures. Therefore, the proposed rank loss is computation-free for inference.

2.2. Misalignment Problem in Visual Tracking

Most of advanced trackers process classification and localization separately, ignoring the misalignment problem between them. Some researchers adopt feature alignment strategies to perform scale-aware correlation. For example, Ocean [59] introduces deformable convolution [10] and SAM [57] utilizes spatial transformer networks. Besides them, many trackers perform re-detection to achieve more accurate localization [40, 42]. However, they usually utilize pool operation and design complicated re-detection mechanism, resulting in high complexity. SiamR-CR [35] and SiamLTR [38] adds an additional branch to evaluate the localization prediction, aiming to achieving localization-sensitive proposal selection criterion. Different from them, we propose an IoU-ranking loss to facilitate back-propagation, which aims to align the confidence score with the associated IoU. It is worth mentioning that our tracker does not need to add any extra network architecture or design any new tracking protocol. So it is totally cost-free at the inference stage.

2.3. Ranking Algorithms

Learning to rank has been widely used in NLP tasks such as recommendation systems, and aims to optimize the ranking of sample lists. Recently, some researchers consider ranking optimization in visual object detection. For example, AP-loss [6] is proposed to optimize the average precision metric of classification directly. DR loss [36] abandons original classification loss and optimizes the ranking of foreground and background distributions. RankDetNet [31] employs two ranking constraints for classification and localization, respectively. However, to our knowledge, learning to rank has not been implemented to visual tracking. Although our ranking-based optimization shares partial similarity with the above methods, the motivation and technical details are quite different: (1) The optimization tasks are different. Ranking strategies in object detection aim to learn class information for the task-special object detection task while our ranking optimization is tailored to enhance similarity measurement utilized in the class-agnostic tracking task. (2) The implementations are different. Our ranking optimization serves as an additional constraint on the original loss, instead of replacing the original ones. (3) The purposes of ranking are different. Ranking strategies utilized in object detection expect that the inter-class variance is large while the intra-class variance is small, which means the samples which belong to the same class should be similar in the classification space. On the contrary, our classifi-

cation ranking loss is used to enlarge the intra-class distance as the target and hard distractors always have the same class.

3. Method

In this section, we will introduce the proposed ranking-based optimization(RBO) based on the Siamese network based trackers. Figure 2 shows the overall pipeline. Firstly, we will briefly review the Siamese trackers in Section 3.1. Then, we will present the proposed RBO in the following sections.

3.1. Revisiting Siamese Trackers

The standard Siamese trackers take an exemplar image z and a search image x as input. The image z points out the concerned target in the first frame, and the trackers are required to locate the target in the search region x in subsequent video frames. The two images are fed into a shared backbone network to generate feature maps $\boldsymbol{F}_z \in \mathbb{R}^{H_z \times W_z \times C}$ and $\boldsymbol{F}_x \in \mathbb{R}^{H_x \times W_x \times C}$, respectively. Then a matching network φ is applied to process $\boldsymbol{F}_z$ and $\boldsymbol{F}_x$ to obtain the similarity feature map $\boldsymbol{S}$ as

$$\boldsymbol{S} = \varphi\left(\boldsymbol{F}_z, \boldsymbol{F}_x\right) \tag{1}$$

Many popular Siamese trackers define φ as depth-wise cross-correlation(DW-Corr) [8, 13, 19, 26, 50, 54]. Recently, inspired by the video object segmentation [34], many researchers take pixel-wise correlation methods(PW-Corr) [17, 18, 29], which are variants of non-local attention [48], as the matching network φ of the tracking task. In this paper, we introduce a simplified version of PW-Corr [17] as

$$w_{ij} = \frac{\exp\left[\left(\boldsymbol{F}_z^i \odot \boldsymbol{F}_x^j\right)/\sqrt{C}\right]}{\sum_{\forall k} \exp\left[\left(\boldsymbol{F}_z^k \odot \boldsymbol{F}_x^j\right)/\sqrt{C}\right]} \tag{2}$$

where $\boldsymbol{F}_z$ and $\boldsymbol{F}_x$ are reshaped into the size of $H_z W_z \times C$ and $C \times H_x W_x$, respectively, and i and j are the indices of each pixel on $\boldsymbol{F}_z$ and $\boldsymbol{F}_x$, respectively. The symbol $\odot$ denotes the dot-product operation. Then we obtain a similarity matrix $w \in \mathbb{R}^{H_z W_z \times H_x W_x}$. The similarity feature map $\boldsymbol{S}$ is calculated as

$$\boldsymbol{S} = \text{concat}\left(\boldsymbol{F}_x, \left(\boldsymbol{F}^z\right)^T \otimes w\right) \tag{3}$$

where $\text{concat}()$ represents matrix concatenation, and $\otimes$ denotes matrix multiplication. Then, the similarity feature map $\boldsymbol{S}$ is fed into the RPN head which consists of classification module θ_{cls} and localization module θ_{loc}. The RPN head could be anchor-based [26, 27] or anchor-free [8, 14]. We can obtain the classification map $\boldsymbol{A}_{cls}$ and the regression map $\boldsymbol{A}_{loc}$ as

$$\boldsymbol{A}_{cls} = \theta_{cls}\left(\boldsymbol{S}\right), \quad \boldsymbol{A}_{loc} = \theta_{loc}\left(\boldsymbol{S}\right) \tag{4}$$

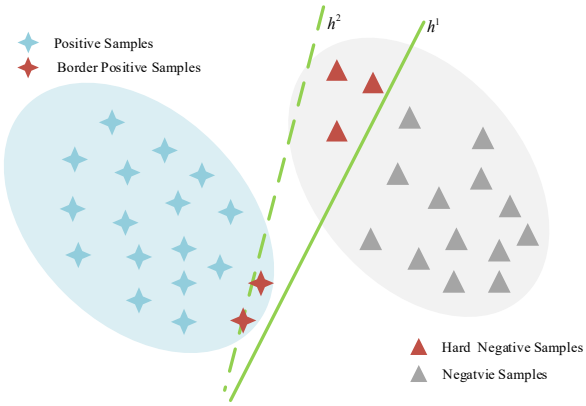

Figure 3. Illustration of binary classification without or with the proposed classification ranking loss. h^1(solid green line) and h^2(dotted green line) are the corresponding decision hyperplanes.

$\boldsymbol{A}_{cls}$ aims to identify the foreground proposals from background ones while $\boldsymbol{A}_{reg}$ regresses the target bounding box. The standard loss of Siamese trackers is defined as

$$\begin{aligned}\mathcal{L}_{rpn} = &\frac{1}{N_{pos}} \sum_{i \in \mathcal{A}_{pos}} \mathcal{L}_{cls}\left(\boldsymbol{A}_{cls}^i, \boldsymbol{Y}_{cls}^i\right) + \mathcal{L}_{loc}\left(\boldsymbol{A}_{loc}^i, \boldsymbol{Y}_{loc}^i\right) \\ &+ \frac{1}{N_{neg}} \sum_{i \in \mathcal{A}_{neg}} \mathcal{L}_{cls}\left(\boldsymbol{A}_{cls}^i, \boldsymbol{Y}_{cls}^i\right)\end{aligned} \tag{5}$$

where N_{pos} and N_{neg} are the numbers of the positive sample set $\mathcal{A}_{pos}$ and the negative sample set $\mathcal{A}_{neg}$, respectively. $\boldsymbol{Y}_{cls}$ and $\boldsymbol{Y}_{loc}$ denote the classification and regression labels, respectively. $\mathcal{L}_{cls}$ is usually the cross entropy loss and $\mathcal{L}_{loc}$ is the commonly used smooth L_1 loss or IoU loss.

We can observe two limitations from Eq. 5. Firstly, the classification branch processes the positive and negative samples separately and does not explore the relationship between them. Secondly, the classification and localization branches are trained with independent objective functions without direct interactions between them, which may yield prediction inconsistency between the classification and localization.

3.2. Classification Ranking Loss

As discussed above, most Siamese-based trackers achieve binary classification via cross entropy loss which can ensure that most samples could be classified correctly. However, as shown in Figure 3, some hard negative samples may cross the decision hyperplane and fool the classifier. In the tracking task, as long as the classification score of one negative sample is larger than those of all positive samples, tracking failure occurs. Hence, false positive classification severely hampers the robustness of trackers.

To alleviate this issue, we propose a classification ranking loss to enlarge the foreground-background classification

decision margin. In particular, we first train the classifier, which is supervised by cross entropy loss. Then we sort all the negative samples by their predicted object confidence scores. The negative samples whose confidence scores are lower than τ_{neg}, e.g., 0.5, are filtered out. The rest ones constitute the hard negative sample set $\{p_{j_-}\}_{j_-}^{n_-}$, where n_- is the number of negative samples, p_{j_-} denotes the object confidence score of sample j_-. Analogously, for the positive samples, we keep all of them to obtain the positive set $\{p_{j_+}\}_{j_+}^{n_+}$. As for the next ranking optimization, we do not employ a point-wise comparison between the set $\{p_{j_-}\}_{j_-}^{n_-}$ and $\{p_{j_+}\}_{j_+}^{n_+}$ due to two reasons. Firstly, the time complexity of the point-wise comparison is equal to $\mathcal{O}(n_-n_+)$, which is expensive for training. Secondly, it is not necessary that each positive sample should rank higher than all negative samples, as some low-confidence positive samples, which are located at the classification borderlines, could be ignored to some extent. Moreover, as long as one positive sample ranks higher than the hard negative ones, the tracker can select the right candidate as the tracking target. In light of the above considerations, we rank the expectations of training samples to enlarge the foreground-background classification margin while the time complexity can be significantly reduced to $\mathcal{O}(1)$. The expectation of hard negative and positive samples are defined as

$$\begin{aligned} P_- &= \sum_{j_-}^{n_-} w_{j_-} p_{j_-}, \\ P_+ &= \sum_{j_+}^{n_+} w_{j_+} p_{j_+}, \end{aligned} \tag{6}$$

where w_{j_-} denotes the expectation weight of sample j_-, w_{j_-} is normalized by the SoftMax function as

$$w_{j_-} = \frac{\exp(p_{j_-})}{\sum_{j_-}^{n_-} \exp(p_{j_-})} \tag{7}$$

Differently, the positive weight w_{j_+} is set as $\frac{1}{n_+}$ since we want to preserve the positive classification distribution. We adopt the logistic loss to rank the expectations P_- and P_+ as

$$\mathcal{L}_{\text{rank-cls}} = \frac{1}{\beta} \log(1 + \exp(\beta \cdot (P_- - P_+ + \alpha)) \tag{8}$$

where β controls the loss value, and α is a ranking margin. Specifically, if there are no hard negative samples in an image, we will skip this image. As shown in Figure 3, with the supervision of $\mathcal{L}_{\text{rank-cls}}$, the decision hyperplane is adjusted from h^1 to h^2 and the hard negative samples are placed at the negative side successfully. Note that some border positive samples may be located at the negative side of the decision hyperplane, which is acceptable for the single object tracking task as we just need one positive sample to represent the tracked target. In addition, Figure 4 further visualizes the effectiveness of the proposed classification ranking loss, where high responses only reflect on the concerned target while the distractors are suppressed significantly.

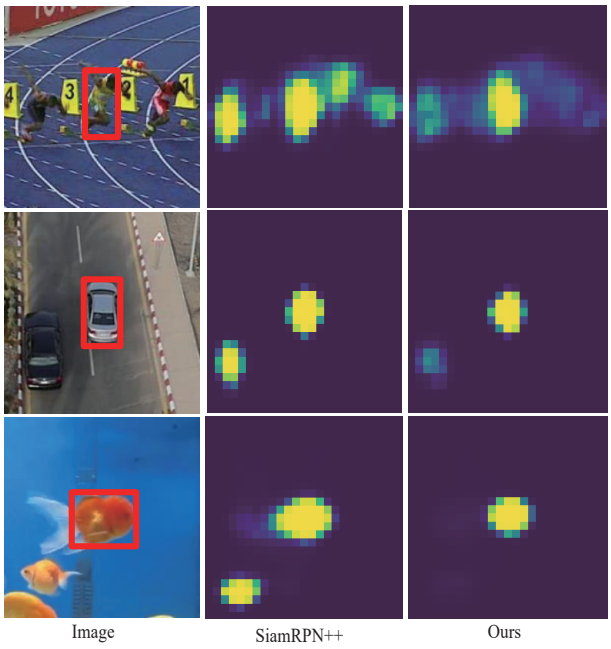

Figure 4. Confidence maps of the target object (red box) estimated by SiamRPN++ and Ours(SiamRPN++ supervised by the proposed classification ranking loss). The model predicted by SiamRPN++ struggles to distinguish the target from distractor objects in the background. In contrast, the proposed ranking loss provides superior discriminative power for SiamRPN++, and can suppress the confidence of distractors significantly.

3.3. IoU-Guided Ranking Loss

To tackle the mismatch between classification confidence and localization, we propose an IoU-guided ranking loss to harmonize the optimization of classification and regression branches. More specifically, the proposed loss aims to align the confidence scores of two positive samples with their associated IoU, and can encourage one positive sample with a larger IoU to rank higher than the other one with a smaller IoU. To this end, for the positive sample $i, j \in \mathcal{A}_{pos}$, the ranking constraints are organized in a pair-wise manner as

$$\begin{aligned} p_i > p_j, \quad s.t.\ v_i^{iou} > v_j^{iou} \\ v_i^{iou} > v_j^{iou}, \quad s.t.\ p_i > p_j \end{aligned} \tag{9}$$

where p_i and p_j indicate the foreground confidence scores of positive samples i, j, respectively, v_i^{iou} and v_j^{iou} denote the predicted IoU values with the ground-truth of samples i

and j. Note that our ranking constraints are modified from RankDetNet [31], and their difference will be analyzed in Section 4.2. Then the IoU-guided ranking loss is defined as

$$L_{\text{rank-iou}} = \frac{1}{N_{pos}} \sum_{i,j \in \mathcal{A}_{pos}, v_i^{iou} > v_j^{iou}} exp(-\gamma \cdot (p_i - p_j)) + \frac{1}{N_{pos}} \sum_{i,j \in \mathcal{A}_{pos}, p_i > p_j} exp(-\gamma \cdot (v_i^{iou} - v_j^{iou})) \tag{10}$$

where $\gamma > 0$ is a hyper-parameter to control the loss value. During the back-propagation optimization process, if $v_i^{iou} > v_j^{iou}$, we will optimize p_i and p_j to make p_i to rank higher than p_j; if $p_i > p_j$, following [31], we will freeze v_j^{iou} and only optimize v_i^{iou} to achieve the expected ranking. If v_j^{iou} is not frozen, the loss could drop by decreasing v_j^{iou}, which would hamper regression optimization.

The proposed IoU-guided ranking loss can narrow the gap between the classification and regression branches by aligning classification scores with the associated IoUs. Thereby well-localized prediction could be represented by high classification confidence.

3.4. Ranking-Based Trackers

Advanced Siamese trackers employ different backbone networks, correlation ways and RPN heads. Since our proposed ranking-based optimization(RBO) aims to facilitate classification and regression optimization, RBO is not sensitive to network architectures. For convenience, we adopt ResNet-50 [21] as backbone network, and integrate the proposed RBO into SiamRPN++(DW-Corr and anchor-based head) [26] and SiamBAN(DW-Corr and anchor-free head) [8], obtaining SiamRPN++-RBO and SiamBAN-RBO, respectively. Moreover, we craft a new version named as SiamPW-RBO by replacing DW-Corr with the introduced PW-Corr in Section 3.1 on the basis of SiamBAN-RBO.

As shown in Figure 2, the proposed two ranking losses can be optimized together with the original loss adopted in Siamese trackers. We empirically combine the original loss $\mathcal{L}_{\text{RPN}}$ with the proposed $\mathcal{L}_{\text{rank-cls}}$ and $\mathcal{L}_{\text{rank-iou}}$ with the weights of 1:0.5:0.25, which benefits a stable offline training. Since the proposed RBO only involves offline training, it does not introduce any extra computation cost at the inference stage.

4. Experiments

Our trackers are implemented using the Pytorch tracking platform PySOT and trained on four NVIDIA GTX 1080Ti GPUs.

Implementation Details. For a fair comparison, we follow the same training protocols(datasets and training hyper-parameters) defined in PySOT for SiamRPN++-RBO, SiamBAN-RBO and SiamPW-RBO. For both training and inference, template patches are resized to the size of 127×127 pixels and the search regions are cropped to 255×255 pixels. The ranking margin α is set as 0.5 according to the analysis of [36]. In order to achieve a stable training, β and γ are set as 4 and 3, respectively, for all experiments.

Evaluation Datasets and Metrics. We use seven tracking benchmarks, including OTB100 [49], UAV123 [33], NFS30 [23], TC128 [28], GOT-10k [22], VOT2016 [24] and LaSOT [15] for the tracking performance evaluation. For VOT2016 [24], we adopt the accuracy(A), robustness(R) and Expected Average Overlap(EAO) metrics. For GOT-10k [22], the trackers are evaluated on its online server, which employs the average overlap (AO) and success rate (SR) metrics. For other datasets, we adopt distance precision(DP) at the 20 pixels and area-under-curve (AUC) score of overlap success plots for evaluation.

4.1. Comparison with State-of-the-art Trackers

OTB100 [49]. We validate our proposed trackers on the OTB100 dataset [49], which consists of 100 fully annotated sequences. As shown in Table 1, our proposed SiamRPN++-RBO, SiamBAN-RBO and SiamPW-RBO achieve the AUC scores of 69.9%, 70.2% and 69.8%, respectively. Compared with the recent proposed Siamese trackers such as SiamRN [9] and SiamGAT [18], our three trackers achieve better or competitive performance against them.

TC128 [28]. For further evaluation, we report tracking results on the TC128 dataset consisting of 128 color sequences. As shown in Table 1, the SiamBAN-RBO and SiamRPN++-RBO outperform existing state-of-the-art Siamese trackers such as SiamGAT [18]. Moreover, SiamRPN++-RBO significantly boosts the AUC score of the baseline SiamRPN++ from 57.3% to 61.9%.

UAV123 [33]. UAV123 dataset contains 123 low-altitude aerial videos captured from a UAV. This dataset has numerous sequences with partial or full occlusions and drastic deformation. As shown in Table 1, our SiamRPN++-RBO obtains a success (AUC) score of 0.643, which significantly outperforms the baseline SiamRPN++ with a large margin. The significant improvement also happens to the SiamBAN-RBO version. This is because the RBO method can increase the difference between representations of the target and background, which contributes to distinguishing the target from distractors.

NFS30 [23]. We conduct experiments on NFS dataset(30 FPS version) [23], which provides 100 challenging videos with fast-moving objects. As shown in Table 1, our three trackers steadily rank top three and outperform recent Siamese trackers like SiamGAT [18].

GOT-10k [22]. The recently released GOT-10k dataset provides a large-scale and high-diversity benchmark where

	SiamRPN++ [26]	SiamBAN [8]	SiamCAR [19]	Ocean [59]	CLNet [13]	CGACD [14]	SiamRN [9]	SiamRPN++ -ACM [20]	SiamBAN -ACM [20]	SiamGAT [18]	**SiamRPN++ -RBO**	**SiamBAN -RBO**	**SiamPW -RBO**
OTB100 [49]	69.6	69.6	65.7	67.2	65.7	71.3	70.1	71.2	72.0	71.0	69.9	70.1	69.8
TC128 [28]	57.3	58.4	57.8	55.1	56.4	60.5	-	-	-	58.5	61.9	61.2	59.3
UAV123 [33]	61.3	61.4	61.4	59.2	63.3	63.3	64.3	63.4	64.6	64.5	64.3	64.1	64.5
NFS30 [23]	50.2	59.0	53.3	51.8	54.3	55.4	-	-	-	56.7	59.6	61.3	60.1

Table 1. Comparison results on on the OTB100 [49], TC128 [28], UAV123 [33] and NFS30 [23] datasets in terms of AUC score. Red, blue and green indicate top three results. Ocean is the offline version.

	SiamRPN++ [26]	Ocean [59]	D3s [32]	SiamFC++ [50]	SiamBAN [8]	SiamCAR [19]	KYS [3]	STMTrack [17]	SiamGAT [18]	**SiamRPN++ -RBO**	**SiamBAN -RBO**	**SiamPW -RBO**
AO(↑)	51.7	59.2	59.7	59.5	57.9	56.9	63.6	64.2	62.7	60.2	60.8	64.4
$SR_{0.5}$(↑)	61.5	69.5	67.6	69.5	68.4	67.0	75.1	73.7	74.3	71.8	72.2	76.7
$SR_{0.75}$(↑)	32.9	47.9	46.2	47.3	45.7	41.5	51.5	57.9	48.8	44.6	46.8	50.9

Table 2. Comparison results on the GOT-10k [22] test set in terms of average overlap (AO), and success rates (SR) at the overlap thresholds of 0.5 and 0.75.

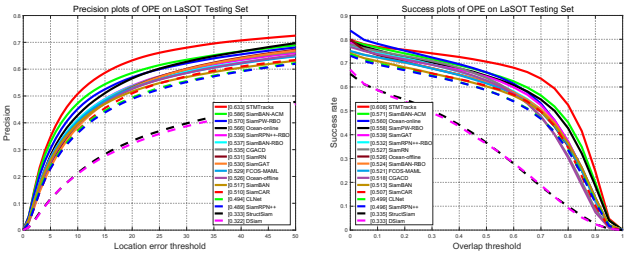

Figure 5. Precision and success plots on the LaSOT [15] *testing* dataset.

the *training* and *testing* subsets have no overlapping. Following its protocol, we just use its *training* subset to train our models and do evaluation on its *testing* set. Table 2 shows the results in terms of average overlap (AO) and success rate (SR) metrics with the overlap thresholds of 0.5 and 0.75. SiamPW-RBO slightly outperforms other top-performing trackers such as memory-based STMTrack [17] in term of AO metric. Compared with SiamRPN++, our SiamRPN++-RBO obtains significant improvements of 8.5% in AO, 10.3% in $SR_{0.5}$ as well as 11.7% in $SR_{0.75}$. The improvement is partially attributed to the fact that the proposed IoU-guided ranking loss can enable trackers to estimate the target states more accurately and alleviate the issue of error accumulation.

VOT2016 [24]. We compare our trackers on VOT2016 in Table 3. VOT2016 contains 60 challenging sequences and ranks the trackers by EAO. As Table 3 reports, our proposed SiamRPN++-RBO outperforms the baseline SiamRPN++ with an absolute gain of 4.9% in EAO. Analogously, SiamBAN-RBO achieves favorable performance against SiamBAN with an EAO of 0.543.

LaSOT [15]. LaSOT is also a large-scale, high-quality dataset. Its *testing* set contains 280 test sequences with an average length of 2506 frames, and is often used to evaluate the long-term tracking performance. As Figure 5 depicts, Our SiamRPN++-RBO and SiamBAN-RBO improve the AUC scores of baselines with the gains of 3.6% and 1.1%, respectively.

Tracker	A(↑)	R(↓)	EAO(↑)
SiamRPN [27]	0.578	0.312	0.337
Da-SiamRPN [60]	0.596	0.266	0.364
SiamDW [58]	0.580	0.240	0.371
SiamMask-*Opt* [47]	0.670	0.230	0.442
UpdateNet [56]	0.610	0.206	0.481
SiamRPN++ [26]	0.642	0.196	0.463
Ocean [59]	0.625	0.158	0.486
ROAM++ [51]	0.559	0.174	0.441
SiamBAN [8]	0.632	0.149	0.502
D3s [32]	0.667	0.158	0.499
SiamRPN++-ACM [20]	0.666	0.144	0.501
SiamBAN-ACM [20]	0.647	0.098	0.549
SiamAttn [54]	0.680	0.140	0.537
SiamRPN++-RBO**(Ours)**	0.635	0.140	0.512
SiamBAN-RBO**(Ours)**	0.629	0.112	0.543
SiamPW-RBO**(Ours)**	0.617	0.098	0.531

Table 3. Comparison with state-of-the-art trackers on the VOT2016 dataset [24] in terms of accuracy(A), robustness(R) and expect average overlap(EAO).

4.2. Ablation Study

In this section, we perform extensive analysis of the proposed three versions on the combined dataset containing the entire TC128 [28], NFS30 [23] and UAV123 [33] datasets. This pooled dataset contains 352 diverse sequences to enable thorough analysis. The AUC score is adopted for evaluation.

Ranking-based Optimization(RBO). We perform a thorough ablation study on each component of RBO. As Table 4 shows, classification ranking loss(CR) improve the AUC scores of SiamRPN++ and SiamBAN by 2.70% and 1.54%, respectively. The significant gains demonstrate that

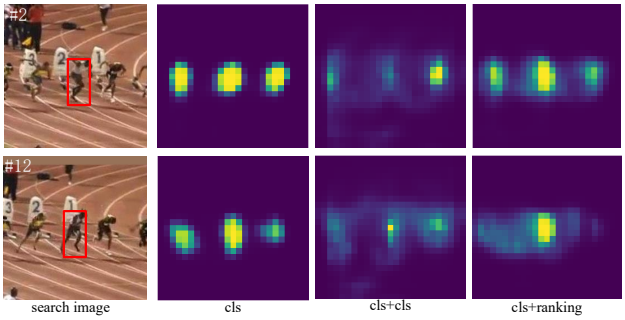

Figure 6. Visualization of confidence maps. From left to right: search image, confidence maps with single-stage classification loss, two-stage classification loss and the classification loss+the proposed ranking loss, respectively.

CR	IGR-ori	IGR	SiamRPN++ [26]	SiamBAN [8]
			57.93	59.61
✓			60.63	61.15
✓	✓		60.87	60.92
✓		✓	62.08	62.24

Table 4. Ablation analysis of the proposed ranking-based optimization, consisting of classification ranking(CR), original IoU-guided ranking [31] (IGR-ori) and our IoU-guided ranking(IGR) losses on the combined TC128, NFS30 and UAV123 datasets.

CR, which models the relationship between the positive samples and hard negative samples, could help to learn a more discriminative classifier. Furthermore, when the trackers are equipped with the IoU-guided ranking loss(IGR), IGR brings the gains of 1.45%(from 60.63% to 62.08%) and 1.09% from (61.15% to 62.24%) for the SiamRPN++ and SiamBAN versions, respectively.

The Strategy of Hard Negative Mining. We test a regular hard negative mining strategy, i.e., employing two-stage cross entropy losses, among which the first one selects hard negative samples and the second one aims to optimize these hard samples. As shown in Figure 6, unfortunately, the real concerned target is suppressed along with the non-target regions, and the classifier still struggles to distinguish the target from the distractors. On the contrary, the proposed method, i.e., cross entropy loss+ranking loss, not only highlights the concerned target, but also suppresses the hard distractors, which confirms that our CR is effective in improving the discrimination of the classifier.

Difference with RankDetNet [31]. As shown in Table 4, the original IoU-guided ranking loss in [31], i.e., $\mathcal{L} = \mathcal{L}_{\text{rank}}\left(-\alpha \cdot (p_i - p_j)\left(v_i^{iou} - v_j^{iou}\right)\right)$, fails to boost our method. We argue that it is not easy for $\mathcal{L}$ to optimize the four variables(p_i, p_j, v_i^{iou}, v_j^{iou}) together since the relationship among the four variables is missing and they may be updated along sub-optimal directions. Differently, in our modified loss(Eq. 10), the joint optimization is divided into two subtasks, and there are only two variables to optimize under the explicit constraint(Eq. 9) in each iteration. As Table 4 shows, our modified loss can further lift the performance, which confirms that more strong and explicit supervision may be more suitable for class-agnostic visual tracking training.

Tracker	LaSOT [15]			GOT-10k [22]			UAV123 [33]	
	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$	DP	AUC
TransT	64.9	73.8	69.0	72.3	82.4	68.2	87.4	67.9
TransT+RBO	65.6	74.3	69.7	72.7	82.9	68.7	88.0	68.5

Table 5. Comparison with the TransT [7] and TransT+RBO methods on the LaSOT, GOT-10k and UAV123 datasets.

4.3. Evaluation on the Transformer trackers

To further evaluate the RBO on the Transformer trackers [7, 45, 52, 53], we choose the TransT [7] method for comparison. From Table 5, TransT+RBO version outperforms the TransT on the three datasets. This indicates that although the Transformer can model the relationship of different proposals by attention mechanism, the RBO can still provide additional cues for facilitating offline optimization.

5. Conclusion and Discussion

In this paper, we propose a ranking-based optimization algorithm for Siamese tracking. Firstly, we propose a classification ranking loss, which converts the classification optimization into ranking problems where the positive samples are encouraged to rank higher than hard negative ones. After the involvement of ranking optimization, the tracker can select the top-rank positive sample as the concerned target without being fooled by distractors. Moreover, in order to reconcile consistency of prediction between classification and localization, we propose an IoU-ranking loss to optimize the classification and localization tasks together at the offline training stage, thereby producing the target estimation with co-occurrence of high classification confidence and localization accuracy at the inference.

Limitations. From Table 5, we observe that the performance gain of our RBO on Transformer tracker is inferior to that on Siamese trackers [8, 26]. In the future, we intend to explore more advanced ranking strategies to boost the Transformer based methods further.

Acknowledgements This work was supported in part by Technological Innovation Project of the New Energy and Intelligent Networked Automobile Industry of Anhui Province(Research, Development and Industrialization of Intelligent Cameras), and in part by the Key Science and Technology Project of Anhui Province under Grant 202203f07020002.

References

[1] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016. 1, 2

[2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, pages 6182–6191, 2019. 3

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, pages 205–221. Springer, 2020. 7

[4] Qi Cai, Yingwei Pan, Yu Wang, Jingen Liu, Ting Yao, and Tao Mei. Learning a unified sample weighting network for object detection. In *CVPR*, pages 14173–14182, 2020. 2

[5] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. In *CVPR*, pages 11583–11591, 2020. 2

[6] Kean Chen, Jianguo Li, Weiyao Lin, John See, Ji Wang, Lingyu Duan, Zhibo Chen, Changwei He, and Junni Zou. Towards accurate one-stage object detection with ap-loss. In *CVPR*, pages 5119–5127, 2019. 3

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, pages 8126–8135, 2021. 8

[8] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *CVPR*, pages 6668–6677, 2020. 2, 4, 6, 7, 8

[9] Siyuan Cheng, Bineng Zhong, Guorong Li, Xin Liu, Zhenjun Tang, Xianxian Li, and Jing Wang. Learning to filter: Siamese relation network for robust tracking. In *CVPR*, pages 4421–4431, 2021. 6, 7

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 3

[11] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, pages 4660–4669, 2019. 3

[12] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *ECCV*, pages 459–474, 2018. 2

[13] Xingping Dong, Jianbing Shen, Ling Shao, and Fatih Porikli. Clnet: A compact latent network for fast adjusting siamese trackers. In *ECCV*, pages 378–395, 2020. 4, 7

[14] Fei Du, Peng Liu, Wei Zhao, and Xianglong Tang. Correlation-guided attention for corner detection based visual tracking. In *CVPR*, pages 6836–6845, 2020. 2, 4, 7

[15] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, pages 5374–5383, 2019. 6, 7, 8

[16] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, pages 7952–7961, 2019. 2

[17] Zhihong Fu, Qingjie Liu, Zehua Fu, and Yunhong Wang. Stmtrack: Template-free visual tracking with space-time memory networks. In *CVPR*, pages 13774–13783, 2021. 2, 3, 4, 7

[18] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. Graph attention tracking. In *CVPR*, pages 9543–9552, 2021. 2, 3, 4, 6, 7

[19] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, and Shengyong Chen. Siamcar: Siamese fully convolutional classification and regression for visual tracking. In *CVPR*, pages 6269–6277, 2020. 2, 4, 7

[20] Wencheng Han, Xingping Dong, Fahad Shahbaz Khan, Ling Shao, and Jianbing Shen. Learning to fuse asymmetric feature maps in siamese trackers. In *CVPR*, pages 16570–16580, 2021. 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 6

[22] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE TPAMI*, 43:1562–1577, 2019. 6, 7, 8

[23] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, pages 1125–1134, 2017. 6, 7

[24] Matej Kristan, Aleš Leonardis, Jiři Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomáš Vojir, Gustav Häger, Alan Lukežič, Gustavo Fernández, et al. The visual object tracking vot2016 challenge results. In *ECCVW*, pages 777–823, 2016. 6, 7

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 1

[26] 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*, pages 4282–4291, 2019. 1, 2, 4, 6, 7, 8

[27] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 1, 2, 4, 7

[28] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE TIP*, 24(12):5630–5644, 2015. 6, 7

[29] Bingyan Liao, Chenye Wang, Yayun Wang, Yaonong Wang, and Jun Yin. Pg-net: Pixel to global matching network for visual tracking. In *ECCV*, pages 429–444, 2020. 2, 3, 4

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2

[31] Ji Liu, Dong Li, Rongzhang Zheng, Lu Tian, and Yi Shan. Rankdetnet: Delving into ranking constraints for object detection. In *CVPR*, pages 264–273, 2021. 2, 3, 6, 8

[32] Alan Lukezic, Jiri Matas, and Matej Kristan. D3s-a discriminative single shot segmentation tracker. In *CVPR*, pages 7133–7142, 2020. 7

[33] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, pages 445–461, 2016. 6, 7, 8

[34] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9226–9235, 2019. 4

[35] Jinlong Peng, Zhengkai Jiang, Yueyang Gu, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, and Weiyao Lin. Siamrcr: Reciprocal classification and regression for visual object tracking. In *IJCAI*, pages 952–958, 2021. 3

[36] Qi Qian, Lei Chen, Hao Li, and Rong Jin. Dr loss: Improving object detection by distributional ranking. In *CVPR*, pages 12164–12172, 2020. 3, 6

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 1

[38] Feng Tang and Qiang Ling. Learning to rank proposals for siamese visual tracking. *IEEE TIP*, 30:8785–8796, 2021. 3

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 1, 3

[40] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *CVPR*, pages 6578–6588, 2020. 3

[41] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *CVPR*, pages 6288–6297, 2020. 3

[42] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. Spm-tracker: Series-parallel matching for real-time visual object tracking. In *CVPR*, pages 3643–3652, 2019. 2, 3

[43] Keyang Wang and Lei Zhang. Reconcile prediction consistency for balanced object detection. In *ICCV*, pages 3631–3640, 2021. 2

[44] Ning Wang, Wengang Zhou, Guojun Qi, and Houqiang Li. Post: Policy-based switch tracking. In *AAAI*, volume 34, pages 12184–12191, 2020. 2

[45] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, pages 1571–1580, 2021. 8

[46] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *CVPR*, pages 4854–4863, 2018. 1, 2

[47] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019. 7

[48] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 3, 4

[49] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE TPAMI*, 37(9):1834–1848, 2015. 6, 7

[50] Yinda Xu, Zeyu Wang, Zuoxin Li, Ye Yuan, and Gang Yu. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *AAAI*, pages 12549–12556, 2020. 1, 2, 4, 7

[51] Tianyu Yang, Pengfei Xu, Runbo Hu, Hua Chai, and Antoni B Chan. Roam: Recurrently optimizing tracking model. In *CVPR*, pages 6718–6727, 2020. 7

[52] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. High-performance discriminative tracking with transformers. In *ICCV*, pages 9856–9865, 2021. 8

[53] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. Learning spatio-temporal transformer for visual tracking. In *ICCV*, pages 10448–10457, 2021. 8

[54] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *CVPR*, pages 6728–6737, 2020. 3, 4, 7

[55] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *CVPR*, pages 8514–8523, 2021. 2

[56] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Learning the model update for siamese trackers. In *ICCV*, pages 4010–4019, 2019. 3, 7

[57] Mengdan Zhang, Qiang Wang, Junliang Xing, Jin Gao, Peixi Peng, Weiming Hu, and Steve Maybank. Visual tracking via spatially aligned correlation filters network. In *ECCV*, pages 469–485, 2018. 3

[58] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, pages 4591–4600, 2019. 7

[59] Zhipeng Zhang, Houwen Peng, Jianlong Fu, Bing Li, and Weiming Hu. Ocean: Object-aware anchor-free tracking. In *ECCV*, pages 771–787, 2020. 3, 7

[60] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 101–117, 2018. 2, 7