

CRACT: Cascaded Regression-Align-Classification for Robust Tracking

Heng Fan^{1,2} and Haibin Ling²

Abstract—High quality object proposals are crucial in visual tracking algorithms that utilize region proposal network (RPN). Refinement of these proposals, typically by box regression and classification in parallel, has been popularly adopted to boost tracking performance. However, it still meets problems when dealing with complex and dynamic background. Thus motivated, in this paper we introduce an improved proposal refinement module, Cascaded Regression-Align-Classification (CRAC), which yields new state-of-the-art performances on many benchmarks.

First, having observed that the offsets from box regression can serve as guidance for proposal feature refinement, we design CRAC as a cascade of box regression, feature alignment and box classification. The key is to bridge box regression and classification via an alignment step, which leads to more accurate features for proposal classification with improved robustness. To address the variation in object appearance, we introduce an identification-discrimination component for box classification, which leverages offline reliable fine-grained template and online rich background information to distinguish the target from background. Moreover, we present pyramid RoIAlign that benefits CRAC by exploiting both the local and global cues of proposals. During inference, tracking proceeds by ranking all refined proposals and selecting the best one. In experiments on seven benchmarks including OTB-2015, UAV123, NfS, VOT-2018, TrackingNet, GOT-10k and LaSOT, our CRACT exhibits very promising results in comparison with state-of-the-art competitors and runs in real-time at 28 fps.

I. INTRODUCTION

As one of the important problems in computer vision, visual tracking has many applications including self-driving, UAV navigation, *etc.* Despite considerable progress made in recent years, robust tracking remains challenging due to many factors such as occlusion, distractor and so on [1].

In this paper we focus on model-free single object tracking. Specifically, given the target in initial frame, a tracker aims at locating it in all subsequent frames by determining its position and scale. Inspired by the Siamese tracking algorithm [2] and the region proposal network (RPN) [3], SiamRPN [4], [5] formulates tracking as an one-shot inference problem and has attracted great attention owing to its excellent performance in both accuracy and speed. It simultaneously predicts classification results and regression offsets for a set of pre-defined anchors to generate proposals. Encouraged by the success of SiamRPN, improvement has been proposed (*e.g.*, [6], [7]) with an additional refinement process, which further regresses and classifies *in parallel* each proposal (see Figure 1(a)). Particularly, regression is

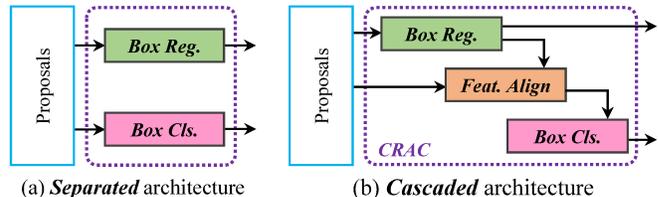


Fig. 1. Different proposal refinement structures: separated box regression and classification in parallel (*e.g.*, [6], [7]) in image (a) and our cascaded regression-align-classification (CRAC) in image (b). *Best viewed in pdf.*

used to adjust the locations and sizes of proposals for better accuracy, and classification to distinguish the target object from background in proposals for better robustness.

Despite improvements achieved, trackers with the above proposal refinement still fail in presence of complex background because of degenerated classification, caused by two problems: (1) In classification task, the features of proposals are directly extracted based on their locations. The inaccuracy in these locations (*e.g.*, due to large scale changes) may contaminate the proposal features (*e.g.*, due to irrelevant background information) and consequently degrades classification results. (2) Background information, which may vary over time and plays a crucial role in distinguishing target from similar objects, is ignored in classification and may hence cause drift to distractors in background.

Contribution. Motivated by aforementioned observations, in this paper we design a new proposal refinement module to improve the robustness of visual tracking.

First, we introduce a novel simple yet effective cascade of regression-align-classification (CRAC) for proposal refinement, which is different than the parallel regression and classification utilized in existing approaches (Figure 1 (a)). This design is motivated by the fact that the offsets from box regression can serve as guidance to sample more accurate proposal features. CRAC consists of three sequential steps, *i.e.*, box regression, feature alignment and box classification, as shown in Figure 1 (b). Specifically, *box regression* aims at further adjusting scales of proposals for better accuracy; *feature alignment* leverages offsets from box regression to better align proposals for improving feature quality; and *box classification* produces refined classification scores for aligned proposals. The key design in CRAC is to connect box regression and classification via an alignment step, instead of separating these two tasks. Such design enables more accurate features of aligned proposals, improving robustness of classification in refinement.

Then, to improve the robustness against background dis-

¹Department of Computer Science and Engineering, University of North Texas, Denton, TX USA. {heng.fan@unt.edu}

²Department of Computer Science, Stony Brook University, Stony Brook, NY USA. {hling@cs.stonybrook.edu}

tractors, we develop an identification-discrimination component in the box classification step of CRAC. Specifically, the identifier learns *offline* a distance measurement and utilizes reliable fine-grained target template to select the proposal most similar to target. The discriminator, drawing inspiration from success in discriminative regression tracking [8], [9], learns *online* a discrete-sampling-based classification model using background and temporal appearance information to suppress similar objects in the proposals. By collaboration of identifier and discriminator, CRAC effectively inhibits distractors in box classification step.

Furthermore, to enhance representation of proposals, we introduce a pyramid RoIAlign (PRoIAlign) module, drawing inspiration from [10], for proposal feature extraction. PRoIAlign is capable of exploiting both local and global cues of proposals, and hence allows CRAC to deal with target deformation and rotation.

We integrate CRAC in the Siamese tracking framework to develop a new tracking algorithm named CRACT (CRAC Tracker). CRACT first extracts a few coarse proposals via a Siamese-style network and then refines each proposal using CRAC. Then the proposal with the highest classification score is selected to be target. In thorough experiments on multiple benchmarks, our CRACT achieves new state-of-the-art results and significantly outperforms its Siamese baselines, while running in real-time.

In summary, we make the following contributions.

- 1) A new cascaded regression-align-classification (CRAC) module is developed for proposal refinement to improve the accuracy and robustness in tracking.
- 2) A novel identification-discrimination component is introduced to leverage offline and online learning of target and background information for handling distractors.
- 3) A pyramid RoIAlign strategy is designed to exploit both local and global cues of proposals for further improving robustness of CRAC.
- 4) A new tracker named CRACT is developed based on the CRAC module, and achieves new state of-the-art results on numerous benchmarks.

II. RELATED WORK

Siamese Tracking. Treating tracking as searching for a region most similar to the initial target template, Siamese network has attracted great attention in tracking. The approach of [11] utilizes a Siamese network to learn a matching function from videos, and then uses it to search for the target object. Despite promising result, this approach runs slowly due to heavy computation. The work of [2] proposes a fully convolutional Siamese network (SiamFC) which efficiently computes the similarity scores of candidate regions. Owing to balanced accuracy and speed, SiamFC has been improved in many follow-ups [12], [13], [4], [5]. Among them, the work of [4] introduces the SiamRPN by combining Siamese network and region proposal network [3] for tracking, achieving more accurate results with faster speed. To improve SiamRPN in dealing with distractors, the

work of [14] leverages more negative training samples for learning a distractor-aware classifier.

Cascade Structure in Tracking. Cascade architecture has been a popular framework for vision tasks (e.g, visual recognition [15] and object detection [16], [17]), and our CRACT also shares this idea for tracking. The approach of [6] presents a two-stage framework in which the proposals generated in the first stage are further identified and refined to choose the best one as the tracking result. The algorithm in [7] suggests a multi-stage framework that cascades multiple RPNs to improve performance of Siamese tracking. The work of [18] introduces cascade architecture in one-stage framework to alleviate misalignment problem in tracking.

Our Approach. In this paper, we regard tracking as a proposal selection task. Our approach is related to but different from SiamRPN [4] which treats tracking as one-shot proposal selection and may suffer from large scale changes and distractors. In contrast, we propose a novel CRAC refinement module to improve proposal selection and achieve better performance. Our method is also relevant to [6], [7] by sharing similar idea of refining proposals. However, unlike in [6], [7] that separately performs regression and classification for refinement, our method takes a cascade structure for refinement. Different from [18] using one-stage architecture for tracking, our trackers is formulated as a two-stage framework.

III. THE PROPOSED APPROACH

We formulate tracking as selecting the best proposal and introduce a novel simple yet effective cascaded regression-align-classification (CRAC) module to refine proposals for such purpose. As shown in Figure 2, our method contains proposal extraction and proposal refinement. In specific, we first use a Siamese region proposal network to filter out most low confident regions and keep only a few initial proposals. Then, each proposal is fed to CRAC module for refinement. During tracking, we rank all refined proposals using the initial and refined classification results, and the proposal with highest score is selected to be the final target. To maintain strong discriminative ability of our tracker, the discriminator in box classification of CRAC is online updated using intermediate results.

A. Proposal Extraction

The goal of proposal extraction is to filter out most negative candidates and retain a few initial proposals similar to target object. This procedure is crucial as one of the proposals from this stage determines the final tracking result. Therefore, it is required to be robust enough to include targets of interest into proposals and to avoid contamination from background. In addition, high efficiency is desired in the proposal extraction. Taking the above reasons into consideration, we leverage Siamese region proposal network, as in [4], [5], [6], [7], for proposal extraction.

The architecture of Siamese RPN contains two branches for target template z and search region x , respectively. As illustrated in Figure 2, using ResNet [19] as backbone, we

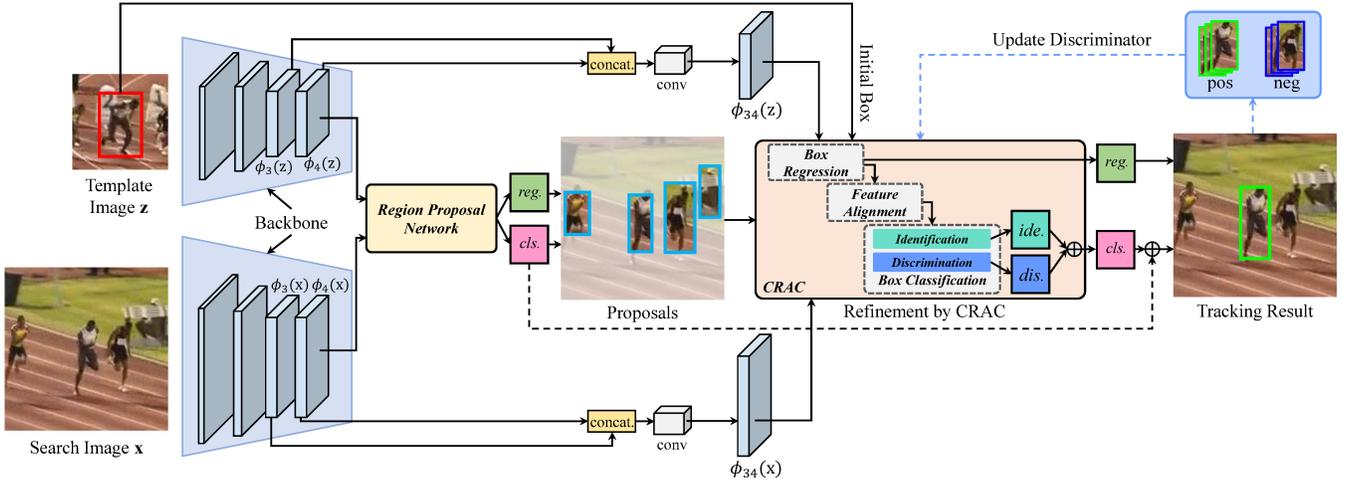


Fig. 2. Illustration of CRACK which first extracts a few coarse proposals (described in section III-A) and then refines each proposal with our cascaded RAC module (described in section III-B). The best proposal is selected based on coarse and refined classification scores to be result. *Best viewed in pdf.*

first extract the features $\phi_4(\mathbf{z})$ and $\phi_4(\mathbf{x})$ after block 4 for \mathbf{z} and \mathbf{x} . Notice that, the feature extraction backbones for \mathbf{z} and \mathbf{x} share the same parameters. Then, $\phi_4(\mathbf{z})$ and $\phi_4(\mathbf{x})$ are fed to RPN, which simultaneously performs classification and regression for predefined anchors on search region. With the classification scores and regression offsets of anchors, we generate N proposals using Non-maximum Suppression (NMS). We represent N proposals as $\{p_i\}_{i=1}^N$, and classification result of p_i is denoted as c_i . The loss ℓ_{RPN} to train Siamese RPN comprises two parts including a cross entropy loss for classification and a smooth L_1 loss [20] for regression. We refer readers to [4], [20] for more details.

B. CRAC for Proposal Refinement

Because the proposals may contain distractors and/or not be good enough to handle large object scale variations, we develop a cascaded regression-align-classification (CRAC) module that refines each coarse proposal by cascading three steps, *i.e.*, *box regression*, *feature alignment* and *box classification*, for better selection. Figure 3 illustrates the architecture of CRAC.

1) *Box Regression*: Since only one-step regression of coarse proposals may not be sufficient to handle object scale changes, we employ an additional box regression in CRAC to further adjust locations and sizes of proposals. In specific, as shown in Figure 3, we first use pyramid RoIAlign (PRoIAlign) module (Section III-C) to extract the feature of each proposal. In order to improve regression accuracy, we employ features from multiple layers. Particularly, we concatenate the features $\phi_4(\mathbf{x})$ and $\phi_3(\mathbf{x})$ after blocks 4 and 3 and use a conv layer to obtain fused feature maps $\phi_{34}(\mathbf{x})$ (Figure 2). Afterwards, the feature f_i of proposal p_i is obtained through PRoIAlign as follows,

$$f_i = \text{PRoIAlign}(\phi_{34}(\mathbf{x}), p_i) \quad (1)$$

As a high-level task, we aim at learning a generic box regression model. Similar to the Siamese tracking [2], [4], we incorporate the target in the first frame as prior information.

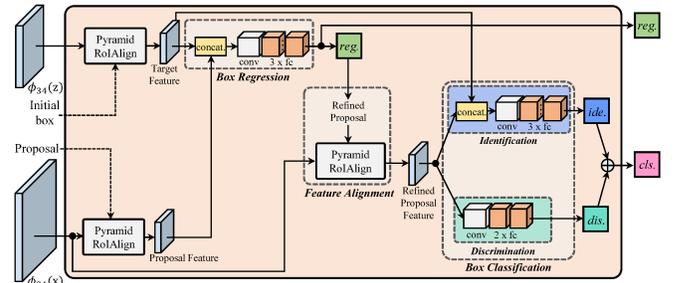


Fig. 3. Illustration of CRAC module. *Best viewed in pdf.*

Likewise, we use multi-level features and obtain the initial target feature as follows,

$$f_{\text{init}} = \text{PRoIAlign}(\phi_{34}(\mathbf{z}), b_1) \quad (2)$$

where $\phi_{34}(\mathbf{z})$ is fused feature maps for target (Figure 2) and b_1 denotes initial object box. Then, the box regression offset r_i of p_i is obtained via

$$r_i = \mathcal{R}(f_i, f_{\text{init}}) \quad (3)$$

where the box regression model \mathcal{R} first concatenates f_i and f_{init} , and then applies a conv layer and three consecutive fc layers to output a 4-dimension vector $r_i = (r_i^x, r_i^y, r_i^w, r_i^h)$. The loss ℓ_{reg} to train the box regression model is smooth L_1 loss [20].

2) *Feature Alignment*: Proposal classification is important, as it greatly affects final proposal selection. Existing refinement method (*e.g.*, [6]) directly extracts proposal features for classification. However, if the locations of proposals are inaccurate, their classification results may be degraded. Thanks to cascade structure of CRAC, we can alleviate this issue by aligning each proposal using offsets from box regression step. By doing so, more accurate proposal features can be used for classification.

In particular, with regression offsets r_i from Eq. (3), we

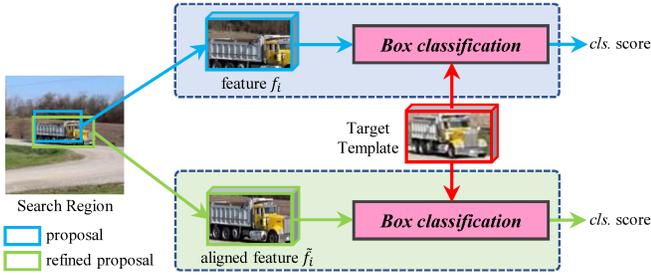


Fig. 4. Comparison of proposal features with and without alignment. We observe that the aligned features are more accurate. *Best viewed in pdf.*

adjust location and size of p_i as follows,

$$\begin{aligned} \tilde{x}_i &= x_i + w_i r_i^x & \tilde{y}_i &= y_i + h_i r_i^y \\ \tilde{w}_i &= w_i \exp(r_i^w) & \tilde{h}_i &= h_i \exp(r_i^h) \end{aligned} \quad (4)$$

where x_i, y_i, w_i, h_i and $\tilde{x}_i, \tilde{y}_i, \tilde{w}_i, \tilde{h}_i$ represent the original and adjusted center coordinates of proposal p_i and its width and height, respectively. With $\tilde{x}_i, \tilde{y}_i, \tilde{w}_i, \tilde{h}_i$, we can obtain the refined proposal \tilde{p}_i for p_i , and extract more accurate feature using \tilde{p}_i via

$$\tilde{f}_i = \text{PRoIAlign}(\phi_{34}(\mathbf{x}), \tilde{p}_i) \quad (5)$$

where \tilde{f}_i represents the aligned feature for p_i . In comparison with f_i , the aligned \tilde{f}_i is more accurate (see Figure 4), which leads to better classification result. In addition, more accurate features can also benefit the training of box classification.

3) *Box Classification*: Since the proposals contain various distractors, a more discriminative classification module is desired in CRAC. Existing methods (*e.g.*, [6], [7]) learn an additional matching sub-network to further classify the proposals for better selection. Owing to more balanced training samples, the classification model in refinement is more discriminative than that for proposal extraction. Despite this, these approaches still fail in presence of hard distractors due to ignorance of background information, which is crucial for distinguishing target from similar objects.

In this work, a joint *identification-discrimination* module is introduced in the box classification step of CRAC. Specifically, the identifier matches *offline* each proposal with reliable target template to find the most similar one. Different from the identifier, the discriminator learns *online* a classification model by exploiting background appearance information to suppress similar objects in proposals. By collaboration of these two components, our method enjoys both reliability of target template to select most similar proposal and the strong discriminative ability to suppress the difficult distractors, leading to robust classification.

Identification. The identifier aims to compute the similarities between proposals and target template. To this end, we leverage a relation network [21] to learn offline a distance measurement between the template and a proposal owing to simplicity and efficiency, similar to [6]. Since the identifier is learned to be generic, no update is required. As an advantage, the identifier will not be contaminated by background, and

thus can resist accumulated errors in discrimination part caused by model update. We compute the identification score \tilde{v}_i for refined proposal \tilde{p}_i as follows,

$$\tilde{v}_i = \mathcal{I}(\tilde{f}_i, f_{\text{init}}) \quad (6)$$

where the identification model \mathcal{I} first concatenates \tilde{f}_i and f_{init} , and then uses a conv layer and three fc layers to obtain a 2-dimension vector \tilde{v}_i , as shown in Figure 3. The loss ℓ_{ide} to train the identification is cross entropy loss.

Discrimination. Different from the identifier, the discriminator focuses on suppressing similar distractors by exploiting background appearance information. For this purpose, we develop an online discrete-sampling-based classifier \mathcal{D} with a light network architecture of one conv and two fc layers, as illustrated in Figure 3. We compute the discrimination score $\tilde{\tau}_i$ for \tilde{p}_i as follows,

$$\tilde{\tau}_i = \mathcal{D}(\tilde{f}_i; \mathbf{w}) \quad (7)$$

where \mathbf{w} denotes the parameters of the discrimination network.

To train discriminator, drawing inspiration from the success of discriminative regression tracking [8], [9], [22], [23], we use the L_2 loss to learn \mathbf{w} as follows,

$$\ell_{\text{dis}} = \sum_{j=1}^M \|\mathcal{D}(X_j; \mathbf{w}) - Y_j\|^2 + \lambda \|\mathbf{w}\|^2 \quad (8)$$

where X_j represents the feature of a training sample, Y_j is a discrete (binary) label, and λ is a regularization parameter. Notice that, unlike identifier trained on image pairs, we generate a set of discrete samples for training discriminator. We utilize the conjugate gradient method in [8] to optimize discrimination network owing to its efficiency. We refer readers to [8] for more details.

It is worth noting that, despite being relevant to discriminative regression tracking [8], [9], [22], [23], our discriminator is different in several aspects: (1) instead of performing classification on a large search region, our method only classifies a few discrete candidate proposals, which is more efficient; (2) the labels of training samples in our method are discrete (binary), which avoids boundary effects by using soft Gaussian labels as in [8], [9], [22], [23]; and (3) because the training samples are discrete, we can easily implement the hard negative mining by focusing more on similar object regions in background.

With Eq. (6) and Eq. (7), we compute the box classification score \tilde{s}_i for refined proposal \tilde{p}_i via

$$\tilde{s}_i = \alpha \cdot \tilde{v}_i^+ + (1 - \alpha) \cdot \tilde{\tau}_i \quad (9)$$

where α is a trade-off parameter and \tilde{v}_i^+ denotes the positive classification score in \tilde{v}_i .

C. Pyramid RoIAlign

Existing refinement approaches like [6] adopt RoIAlign [24] to extract proposal features. Specifically, the features of proposals are usually pooled to a fixed size (*e.g.*, 6×6).

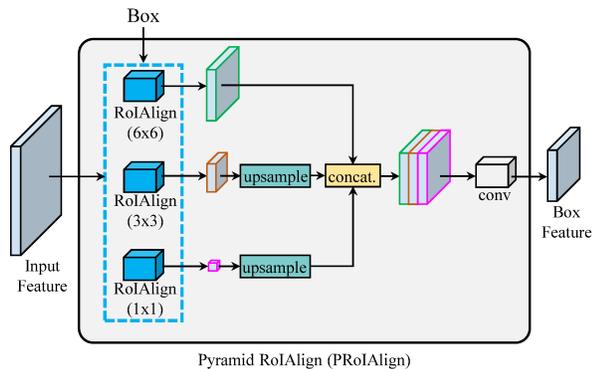


Fig. 5. Illustration of pyramid RoIAlign.

Algorithm 1: Tracking with CRACT

Input: Image sequences $\{\mathbf{I}\}_{t=1}^T$, initial target box b_1 and trained model CRACT;

Output: Tracking result $\{b_t\}_{t=2}^T$;

- 1 Crop target template \mathbf{z} in \mathbf{I}_1 using b_1 ;
 - 2 Extract feature embeddings $\phi_{34}(\mathbf{z})$ and f_{init} for \mathbf{z} ;
 - 3 **for** $t = 2$ to T **do**
 - 4 Crop the search region \mathbf{x} in \mathbf{I}_t using b_{t-1} ;
 - 5 Extract feature embedding $\phi_{34}(\mathbf{x})$ for \mathbf{x} ;
 - 6 Extract proposals $\{p_i\}_{i=1}^N \leftarrow \text{RPN}(\phi_{34}(\mathbf{z}), \phi_{34}(\mathbf{x}))$;
 - 7 Extract features $\{f_i\}_{i=1}^N$ for proposals;
 - 8 Box regression to obtain $\{r_i\}_{i=1}^N$ using Eq. (3);
 - 9 Feature alignment to obtain $\{\tilde{f}_i\}_{i=1}^N$ using Eq. (5);
 - 10 Box classification to obtain $\{\tilde{s}_i\}_{i=1}^N$ using Eq. (9);
 - 11 Select the best proposal to determine the target box b_t using Eq. (10);
 - 12 Collect training samples based on b_t and update the discriminator when necessary;
-

Despite simplicity, such features may be constrained to local target information and therefore sensitive to rotation and deformation. To alleviate this problem, we introduce a pyramid RoIAlign (PROAlign) module, which utilizes multiple RoIAlign operations to extract proposal features at different pooling sizes. For example, for size 1×1 , the proposal features contain global target information. To leverage both local and global cues, pooled features with different sizes are concatenated for fusion to derive more robust local-global proposal features. Figure 5 illustrates the architecture of our PROAlign module. In our implementation, the PROAlign module is designed to have three levels, *i.e.*, 6×6 , 3×3 and 1×1 , for proposal feature extraction.

D. Training and Tracking

Training. The training of CRACT comprises two parts: (1) *offline training* of Siamese RPN, box regression and identifier, and (2) *online training* of discriminator in box classification. The first part is trained using image pairs, and the total training loss $\mathcal{L} = \ell_{\text{rpn}} + \ell_{\text{reg}} + \ell_{\text{ide}}$. Similar to [4], [6], the ratios of anchors are set to $[0.33, 0.5, 1, 2, 3]$ in RPN. The intersection over union (IoU) thresholds to determine anchors as positive (greater than threshold) or negative (less than threshold) are 0.6 and 0.3. We generate up to 64 samples

TABLE I

COMPARISON WITH STATE-OF-THE-ARTS ON OTB-2015 [26]. THE BEST THREE RESULTS HIGHLIGHTED IN RED, GREEN AND BLUE, RESPECTIVELY, THROUGHOUT THE REST OF THE PAPER.

Tracker	Where	PRE Score	SUC Score
MDNet [25]	CVPR'16	0.909	0.678
SiamFC [2]	ECCVW'16	0.771	0.582
ECO [9]	CVPR'17	0.910	0.691
PTAV [27]	ICCV'17	0.849	0.635
SA-Siam [28]	CVPR'18	0.865	0.657
DaSiamRPN [14]	ECCV'18	0.880	0.658
SiamRPN++ [5]	CVPR'19	0.915	0.696
C-RPN [7]	CVPR'19	0.885	0.663
SPM-18 [6]	CVPR'19	0.912	0.701
SiamDW [12]	CVPR'19	0.900	0.670
ATOM [8]	CVPR'19	0.864	0.655
DiMP-50 [29]	ICCV'19	0.900	0.688
SiamBAN [30]	CVPR'20	0.910	0.696
Retina-MAML [31]	CVPR'20	n/a	0.712
SiamAttn [32]	CVPR'20	0.926	0.712
CRACT (Ours)	-	0.936	0.726

from one image pair for RPN training. We choose at most 16 and 32 proposals for box regression and identifier training, respectively. The IoU thresholds to determine the proposals at positive (greater than threshold) or negative (less than threshold) are both 0.5. The second part is online trained during tracking. In particular, we draw 200 positive and 1000 negative samples in the first frame for initial training. The optimization strategy for training and update follows [8] except training samples are discrete.

Tracking by Proposal Selection. We formulate tracking as selecting the best proposal. For each sequence, we extract feature embeddings for target and initialize discriminator. When a new frame arrives, we crop a search region and perform RPN to generate proposals $\{p_i\}_{i=1}^N$, which are refined by CRAC to obtain $\{\tilde{p}_i\}_{i=1}^N$. We rank $\{\tilde{p}_i\}_{i=1}^N$ using coarse and refined classification scores and the target box b is determined by the proposal with the highest score as follows,

$$b = \arg \max_{\tilde{p}_i} (\beta \cdot \tilde{s}_i + (1 - \beta) \cdot \tilde{c}_i) \quad (10)$$

where $\tilde{c}_i = c_i$ and \tilde{s}_i denote respectively coarse and refined scores of \tilde{p}_i , and β is a trade-off parameter. With tracking target box b , we collect n^+ positive and n^- negative samples every K frames to update the discriminator. We leverage short-long update strategy in [25]. Notice that, we only update the two fc layers in the discrimination network. To improve robustness, we use hard negative mining by increasing the number of similar distractors in negative samples. Algorithm 1 summarizes the tracking with CRACT.

IV. EXPERIMENTS

Implementation. We implement CRACT in python using PyTorch [42] on a single GTX 1080 GPU with 8GB memory. We utilize ResNet-18 [19] as backbone and borrow its parameters trained on ImageNet [43]. The number N of proposals during tracking is empirically set to 10. The trade-off parameters α and β are 0.4 and 0.8, respectively. The update interval K for the discriminator is 10. n^+ and n^-

TABLE II
COMPARISON WITH STATE-OF-THE-ART TRACKERS ON UAV123 [33].

Tracker	ECOhc [9]	ECO [9]	SiamRPN [4]	RT-MDNet [34]	DaSiam RPN [14]	ARCF [35]	SiamRPN ++ [5]	ATOM [8]	DiMP-50 [29]	SiamBAN [30]	SiamAttn [32]	CRACT (ours)
Where	CVPR'17	CVPR'17	CVPR'18	ECCV'18	ECCV'18	CVPR'19	CVPR'19	CVPR'19	ICCV'19	CVPR'20	CVPR'20	-
PRE	0.725	0.741	0.748	0.772	0.796	0.670	0.807	0.856	0.858	0.833	0.845	0.860
SUC	0.506	0.525	0.527	0.528	0.586	0.470	0.613	0.642	0.653	0.631	0.650	0.664

TABLE III
COMPARISON WITH STATE-OF-THE-ART TRACKERS ON NFS [36].

Tracker	HCF [37]	HDT [38]	MDNet [25]	SiamFC [2]	ECOhc [9]	ECO [9]	BACF [39]	UPDT [40]	ATOM [8]	DiMP-50 [29]	SiamBAN [30]	CRACT (ours)
Where	ICCV'15	CVPR'16	CVPR'16	ECCVW'16	CVPR'17	CVPR'17	ICCV'17	ECCV'18	CVPR'19	ICCV'19	CVPR'20	-
SUC	0.295	0.403	0.429	0.401	0.459	0.466	0.341	0.542	0.590	0.619	0.594	0.625

TABLE IV
COMPARISON WITH OTHER TRACKERS ON VOT-2018 [41].

Tracker	SiamFC [2]	ECO [9]	SA-Siam [28]	SiamRPN [4]	UPDT [40]	DaSiam RPN [14]	SiamRPN ++ [5]	ATOM [8]	DiMP-50 [29]	SiamBAN [30]	Retina-MAML [31]	CRACT (ours)
Where	ECCVW'16	CVPR'17	CVPR'18	CVPR'18	ECCV'18	ECCV'18	CVPR'19	CVPR'19	ICCV'19	CVPR'20	CVPR'20	-
Acc.	0.500	0.480	0.543	0.588	0.536	0.590	0.600	0.590	0.597	0.597	0.604	0.611
Rob.	0.590	0.280	0.224	0.276	0.184	0.280	0.234	0.204	0.153	0.178	0.159	0.175
EAO	0.188	0.276	0.325	0.384	0.376	0.383	0.414	0.401	0.440	0.452	0.452	0.455

are set to 50 and 200, respectively. The learning rate of the offline training part is 10^{-2} with a decay of 10^{-4} . It is trained end-to-end with SGD by 50 epochs. We apply LaSOT [44], TrackingNet [45], GOT-10k [46] and COCO [47] for offline training, excluding the one under testing. The online training and update of the discriminator utilizes the strategy in [8]. Our tracker runs at 28 frames per second (*fps*). The implementation will be available at <https://hengfan2010.github.io/publications.html>.

A. State-of-the-art Comparison

OTB-2015 [26]. OTB-2015 is a popular tracking benchmark with 100 videos. We compare CRACT with 15 trackers. The comparison is demonstrated in Table I with precision (PRE) and success (SUC) scores using one-pass evaluation (OPE). CRACT achieves the best results with 0.936 PRE score and 0.726 SUC score, outperforming the second best by 1.0% and 1.4%, respectively. Compared with SiamRPN++ with 0.915 PRE score and 0.696 SUC score, we achieve 2.1% and 3.0% gains owing to RAC. Besides, compared to proposal refinement method SPM-18, which can serve as our baseline, with 0.912 PRE score and 0.701 SUC score, CRACT with cascaded refinement shows 2.4% and 2.5% improvements, evidencing effectiveness in boosting tracking robustness.

UAV123 [33]. UAV123 focuses on aerial object tracking and contains 123 videos. We compare CRACT to 11 trackers and the results are displayed in Table II. CRACT obtains the best 0.860 PRE score and 0.664 SUC score, outperforming the second best DiMP-50 with 0.858 PRE score and 0.653 SUC score. In comparison to SiamRPN++ with 0.613 SUC score, we achieve 5.1% absolute gain, which clearly shows the advantage of our proposal refinement.

TABLE V
COMPARISON WITH OTHER TRACKERS ON TRACKINGNET [45].

Tracker	Where	PRE Score	NPRE Score	SUC Score
C-RPN [7]	CVPR'19	0.619	0.746	0.669
SiamRPN++ [5]	CVPR'19	0.694	0.799	0.733
SPM [6]	CVPR'19	0.661	0.778	0.712
ATOM [8]	CVPR'19	0.648	0.771	0.703
DiMP-50 [29]	ICCV'19	n/a	0.801	0.740
Retina-MAML [31]	CVPR'20	n/a	0.786	0.698
SiamAttn [32]	CVPR'20	n/a	0.817	0.752
CRACT (ours)	-	0.724	0.824	0.754

Nfs [36]. Nfs consists of 100 sequences for evaluation on high frame rate videos. We evaluate our approach on 30 *fps* version. Table III demonstrates our result and comparison to 11 trackers. Our CRACT achieves the best result with 0.625 SUC score, which outperforms the second best DiMP-50 with 0.619 SUC score by 0.6%.

VOT-2018 [41]. VOT-2018 contains 60 videos for tracking. We compare CRACT with 11 trackers and Table IV demonstrates the comparison results. Our tracker achieves the best of 0.455 on EAO. Compared to SiamRPN++ which also regards tracking as proposal selection, CRACT obtains a performance gain of 4.1% in term of EAO, which shows the effectiveness of our hierarchical RAC in refining proposals for better selection. Compared to the recent DiMP-50 with 0.440 EAO score, our method achieves 1.5% improvement.

TrackingNet [45]. TrackingNet offers 511 videos for evaluation. Table V shows comparison results of CRACT with 7 state-of-the-art trackers. Our method achieves the best results of 0.724, 0.824 and 0.754 on PRE, NPRES and SUC scores, outperforming recent trackers SiamAttn and DiMP-50.

TABLE VI
COMPARISON RESULTS ON GOT-10K [46].

Tracker	MDNet [25]	SiamFC [2]	SPM [6]	ATOM [8]	DiMP-50 [29]	CRACT (ours)
Where	CVPR'16	ECCVW'16	CVPR'19	CVPR'19	ICCV'19	-
AO	0.299	0.348	0.513	0.556	0.611	0.620
SR _{0.50}	0.303	0.353	0.593	0.634	0.717	0.728
SR _{0.75}	0.099	0.098	0.359	0.402	0.492	0.496

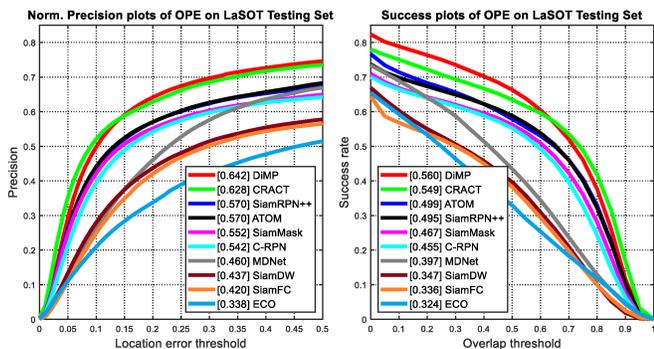


Fig. 6. Comparison with state-of-the-arts on LaSOT [44].

LaSOT [44]. LaSOT is a recent long-term tracking benchmark. We evaluate CRACT under protocol II. Figure 6 shows our results and comparison with 9 state-of-the-arts. CRACT achieves the second best results with 0.628 normalized PRE score and 0.549 SUC score, slightly lower than the 0.642 normalized PRE score and 0.560 SUC score by DiMP-50. Compared with ATOM and SiamRPN++ with 0.499 and 0.495 SUC scores, CRACT shows clear performance gains of 5.0% and 5.4%.

GOT-10k [46]. GOT-10k offers 180 challenging videos for short-term tracking evaluation. We compare CRACT to 5 trackers as displayed in Table VI. CRACT performs the best with 0.620 AO score, outperforming the second best DiMP-50 with 0.611 AO score. Besides, CRACT obtains a significant performance gain of 10.7% compared to SPM.

Figure 7 demonstrates qualitative tracking results on challenging videos. As shown in Figure 7, our CRACT is able of robustly locate the target object in all sequences.

B. Ablation Study

Cascade structure. This paper introduces a novel proposal refinement module with cascade structure. We verify its effectiveness by designing a refinement module with parallel structure by removing feature alignment. Table VII shows results of parallel and cascade refinement. We observe that CRACT with parallel refinement achieves SUC scores of 0.713 and 0.609 on OTB-2015 and NfS. By utilizing cascaded proposal refinement, the results are significantly improved to 0.726 (1.3% gain) and 0.625 (1.6% gain), which clearly evidences the advantage of cascade architecture.

Identification-discrimination. We propose a joint module of discrimination and discrimination in CRAC for proposal classification. In fact, either the identifier or discriminator can be used individually for proposal classification. However,

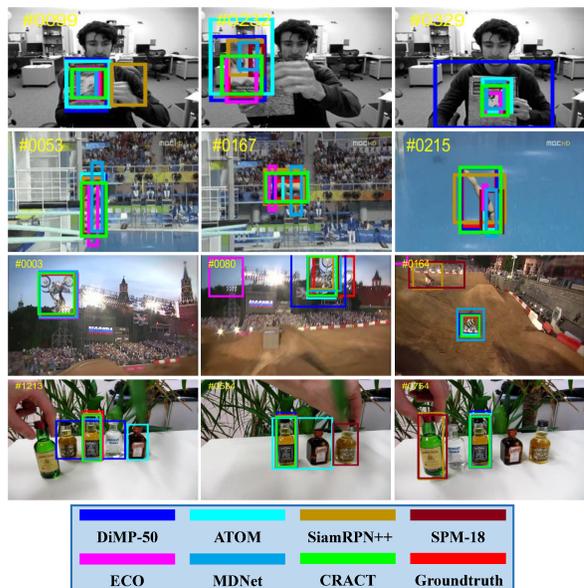


Fig. 7. Qualitative tracking results of CRACT and comparison with six tracking algorithms on four challenging sequences in OTB-2015 (from top to bottom: *Clifbar*, *Diving*, *Motorrolling* and *Liquor*). Our method can robustly locate the target object in different challenges.

TABLE VII
COMPARISON OF SIMULTANEOUS AND HIERARCHICAL REFINEMENT.

	Parallel refinement	Cascaded refinement
SUC on OTB-2015	0.713	0.726
SUC on NfS	0.609	0.625
Speed	31 <i>fps</i>	28 <i>fps</i>

TABLE VIII
COMPARISON (IN SUC) BETWEEN INDIVIDUAL AND JOINT USE OF IDENTIFIER AND DISCRIMINATOR.

	Identifier only	Discriminator only	Joint
OTB-2015	0.715	0.712	0.726
NfS	0.606	0.614	0.625
Speed	43 <i>fps</i>	36 <i>fps</i>	28 <i>fps</i>

TABLE IX
COMPARISON BETWEEN ROIALIGN AND PYRAMID ROIALIGN.

	RoIAlign	PRoIAlign
SUC on OTB-2015	0.719	0.726
SUC on NfS	0.615	0.625
Speed	32 <i>fps</i>	28 <i>fps</i>

each has advantages and disadvantages. The identifier can easily recognize the target from non-semantic distractors using powerful distance measurement. In addition, it avoids the contamination by background owing to no update. Nevertheless, it cannot leverage appearance information. The discriminator works well in suppressing semantic distractors through online learning background information. Nonethe-

less, it has a risk of model contamination caused by update. By collaboration of identifier and discriminator, they can complement each other for better robust proposal selection. We verify the effects of individual and joint use of identifier and discriminator. Table VIII shows the comparison. Using identifier only and discriminator only achieves SUC scores of 0.715 and 0.712 on OTB-215. with joint consideration of them, the performance is significantly boosted to 0.726. Likewise, the best result of 0.625 SUC score is obtained when combining identifier and discriminator.

Pyramid RoIAlign. Different from current tracker [6] using RoIAlign [24] for proposal extraction, we present a simple yet effective PRoIAlign to exploit global and local cues. Table IX shows the results with RoIAlign and our PRoIAlign. We observe that PRoIAlign improves the SUC scores from 0.719 to 0.716 on OTB-2015 and from 0.615 to 0.625 on NfS, respectively, showing the advantage of exploring various cues in performance improvement.

V. CONCLUSION

In this paper, we propose a novel tracker dubbed CRACT for accurate and robust tracking. CRACT first extracts a few coarse proposals and then refines each proposal using the proposed cascaded regression-align-classification module. During inference, the best proposal determined by both coarse and refined classification scores is selected to be the final target. Experiments on seven benchmarks demonstrate its superior performance.

Acknowledgment. This work is supported in part by NSF Grants IIS-2006665 and IIS-1814745.

REFERENCES

- [1] H. Fan, F. Yang, P. Chu, L. Yuan, and H. Ling, "Trackclinic: Diagnosis of challenge factors in visual tracking," in *WACV*, 2021.
- [2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCVW*, 2016.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
- [4] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *CVPR*, 2018.
- [5] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *CVPR*, 2019.
- [6] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "Spm-tracker: Series-parallel matching for real-time visual object tracking," in *CVPR*, 2019.
- [7] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *CVPR*, 2019.
- [8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "Atom: Accurate tracking by overlap maximization," in *CVPR*, 2019.
- [9] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *CVPR*, 2017.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *TPAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [11] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *CVPR*, 2016.
- [12] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *CVPR*, 2019.
- [13] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *ICCV*, 2017.
- [14] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *ECCV*, 2018.
- [15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001.
- [16] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018.
- [17] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting rcnn: On awakening the classification power of faster rcnn," in *ECCV*, 2018.
- [18] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *ECCV*, 2020.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [20] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.
- [21] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *CVPR*, 2018.
- [22] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *ECCV*, 2018.
- [23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *TPAMI*, vol. 37, no. 3, pp. 583–596, 2014.
- [24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.
- [25] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.
- [26] Y. Wu, J. Lim, and M. Yang, "Object tracking benchmark," *TPAMI*, vol. 37, no. 9, pp. 1834–1848, 2015.
- [27] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *ICCV*, 2017.
- [28] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *CVPR*, 2018.
- [29] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *ICCV*, 2019.
- [30] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *CVPR*, 2020.
- [31] G. Wang, C. Luo, X. Sun, Z. Xiong, and W. Zeng, "Tracking by instance detection: A meta-learning approach," in *CVPR*, 2020.
- [32] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable siamese attention networks for visual object tracking," in *CVPR*, 2020.
- [33] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *ECCV*, 2016.
- [34] I. Jung, J. Son, M. Baek, and B. Han, "Real-time mdnet," in *ECCV*, 2018.
- [35] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *CVPR*, 2019.
- [36] H. Kiani Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for speed: A benchmark for higher frame rate object tracking," in *ICCV*, 2017.
- [37] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *ICCV*, 2015.
- [38] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *CVPR*, 2016.
- [39] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, 2017.
- [40] G. Bhat, J. Johnander, M. Danelljan, F. Shahbaz Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *ECCV*, 2018.
- [41] M. Kristan *et al.*, "The sixth visual object tracking vot2018 challenge results," in *ECCVW*, 2018.
- [42] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.
- [43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.
- [44] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *CVPR*, 2019.
- [45] M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "Trackingnet: A large-scale dataset and benchmark for object tracking in the wild," in *ECCV*, 2018.
- [46] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *TPAMI*, 2019.
- [47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.