

- GlobalTrack 1
- Abstract 1
- ▶ Introduction 1
- Related Work 3
- ▼ This Approach 3
 - QG-RPN 3
 - ▶ QG-RCNN 3
 - Cross-query Loss 4
 - Offline Training 4
 - Online Tracking 4
- ▶ Experiments 4
- Conclusions 7

GlobalTrack: A Simple and Strong Baseline for Long-term Tracking

Lianghua Huang^{1,2}, Xin Zhao^{1,2*}, Kaiqi Huang^{1,2,3}

¹CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing, China

²University of Chinese Academy of Sciences, Beijing, China

³CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China

huanglianghua2017@ia.ac.cn, {xzhao, kqhuang}@nlpr.ia.ac.cn

Abstract

A key capability of a long-term tracker is to search for targets in very large areas (typically the entire image) to handle possible target absences or tracking failures. However, currently there is a lack of such a strong baseline for global instance search. In this work, we aim to bridge this gap. Specifically, we propose GlobalTrack, a pure global instance search based tracker that makes no assumption on the temporal consistency of the target's positions and scales. GlobalTrack is developed based on two-stage object detectors, and it is able to perform full-image and multi-scale search of arbitrary instances with only a single query as the guide. We further propose a cross-query loss to improve the robustness of our approach against distractors. With no online learning, no punishment on position or scale changes, no scale smoothing and no trajectory refinement, our pure global instance search based tracker achieves comparable, sometimes much better performance on four large-scale tracking benchmarks (i.e., 52.1% AUC on LaSOT, 63.8% success rate on TLP, 60.3% MaxGM on OxUvA and 75.4% normalized precision on TrackingNet), compared to state-of-the-art approaches that typically require complex post-processing. More importantly, our tracker runs without cumulative errors, i.e., any type of temporary tracking failures will not affect its performance on future frames, making it ideal for long-term tracking. We hope this work will be a strong baseline for long-term tracking and will stimulate future works in this area. Code is available at <https://github.com/huanglianghua/GlobalTrack>.

Given an arbitrary, user-specified target in the first frame, the task of visual tracking is to locate it continuously in successive frames. Visual tracking has been widely used in many fields such as surveillance, augmented reality, robotics and video editing. Over the past decades, significant progress has been made in this area (Wu, Lim, and Yang 2015; Valmadre et al. 2018; Huang, Zhao, and Huang 2018).

Most existing trackers work under a strong temporal consistency assumption that the target's position and scale change smoothly. These approaches typically search within a small window for the target and impose penalties on large position and scale variations to limit the prediction

*Corresponding author

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

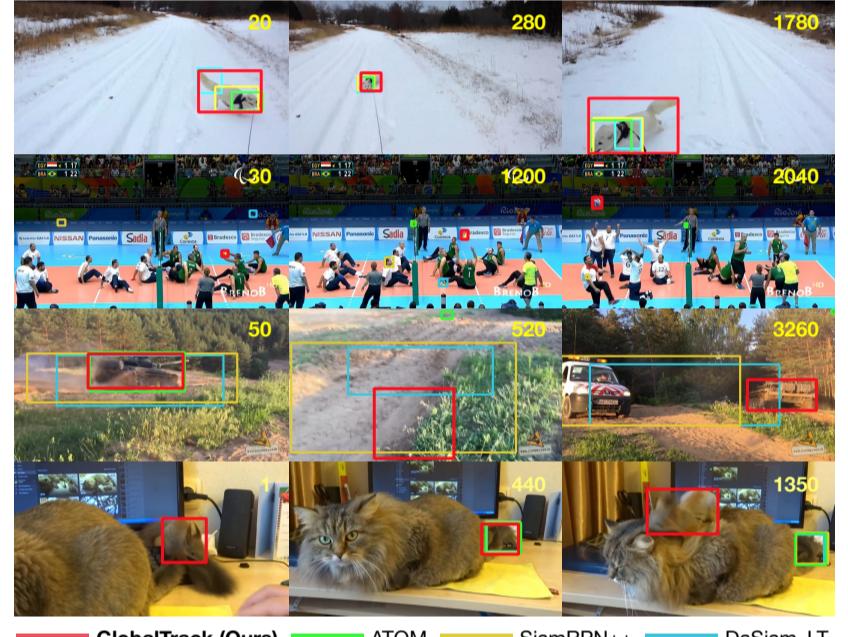


Figure 1: A comparison of our approach with state-of-the-art trackers. ATOM (Danelljan et al. 2019), SiamRPN++ (Li et al. 2019) and DaSiam_LT (Zhu et al. 2018) work under the temporal consistency assumption that the target's state changes smoothly. However, as the figure shows, such assumption does not necessarily hold. Under abrupt motion and temporary target absence, these trackers cannot locate the targets for long. Our approach GlobalTrack, a pure global instance search based tracker, successfully handles these challenges and provides robust tracking results.

space (Bertinetto et al. 2016; Li et al. 2018; Huang, Zhao, and Huang 2019; Danelljan et al. 2019). However, various real-world challenges can break such assumption and cause these approaches to fail. Figure 1 shows typical failure cases of some state-of-the-art trackers, where under abrupt position and scale changes, target absences and temporary tracking failures, none of these approaches is able to consistently locate the targets for long.

In this paper, we propose a baseline tracker that handles these challenges using pure global instance search. The key idea is to remove the locality assumption and enable the tracker to search for the target at arbitrary positions and

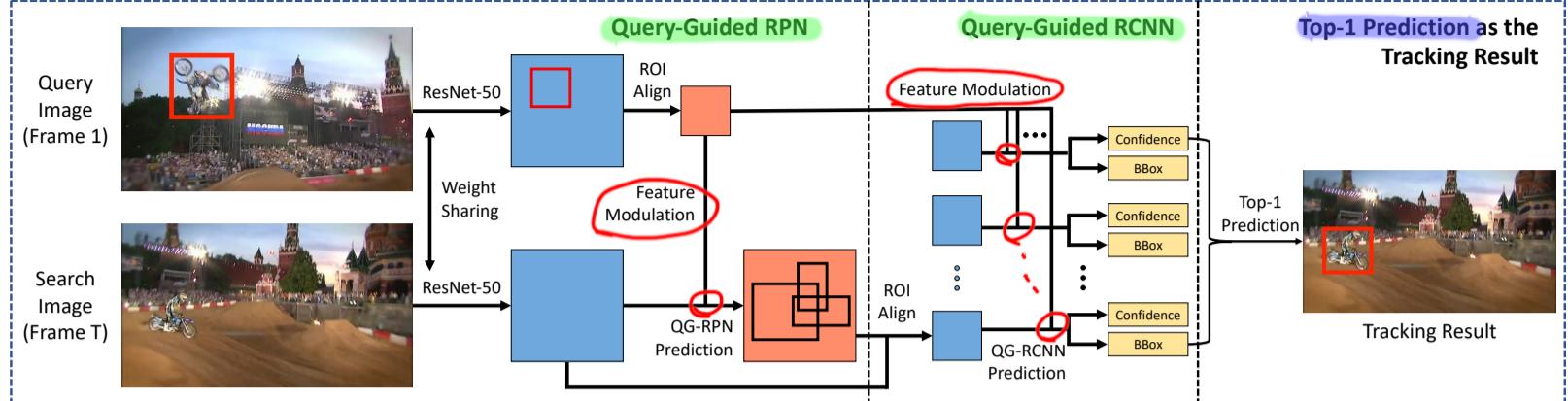


Figure 2: The overall architecture of GlobalTrack. The architecture consists of two submodules: a Query-Guided RPN (QG-RPN) for generating query-specific proposals, and a Query-Guided RCNN (QG-RCNN) for discriminating the proposals and producing the final predictions. In the feature modulation of QG-RPN and QG-RCNN, we encode the correlation between the query and search image features in the backbone and ROI outputs, so as to guide the detector to locate query-specific instances. During tracking, we use the first frame as the query and simply take the top-1 prediction in a frame as the tracking result.

scales, thereby avoiding cumulative errors during tracking. We build such global instance search algorithm based on object detection models, since they are able to perform full-image and multi-scale search of arbitrary sized objects.

Specifically, we propose GlobalTrack, a full-image visual tracker, inspired by the two-stage object detector Faster-RCNN (Girshick 2015). Similar to Faster-RCNN, GlobalTrack consists of two submodules: a query-guided region proposal network (QG-RPN) for generating query-specific object candidates, and a query-guided region convolutional neural network (QG-RCNN) for classifying the candidates and producing the final predictions. The overall architecture of GlobalTrack is visualized in Figure 2. In the feature modulation parts of QG-RPN and QG-RCNN, we encode the correlation between the query and search image features in the backbone and ROI outputs, thereby directing the detector to locate query-specific instances.

During tracking, we take the annotated first frame as the query and search independently in each of the rest frames for the target. We simply take the top-1 prediction of QG-RCNN as the tracking result, without using any further post-processing. While adding additional processes, such as trajectory smoothing, may improve the performance of our approach, we prefer to keep the current model simple and straightforward. In the training phase, we sample frame pairs from video datasets to optimize the model, utilizing the same classification and localization losses as in Faster-RCNN. We further propose a cross-query loss to improve the robustness of GlobalTrack against instance-level distractors, which averages the losses over different queries on a same image to force the model to learn the strong dependency between the queries and the prediction results.

We verify the performance of our approach on four large scale tracking benchmarks: LaSOT (Fan et al. 2019), TrackingNet (Muller et al. 2018), TLP (Moudgil and Gandhi 2018) and OxUvA (Valmadre et al. 2018), where LaSOT, OxUvA and TLP are long-term tracking benchmarks with average video lengths of 2500, 4260 and 13529 frames, respectively. GlobalTrack shows impressive performance

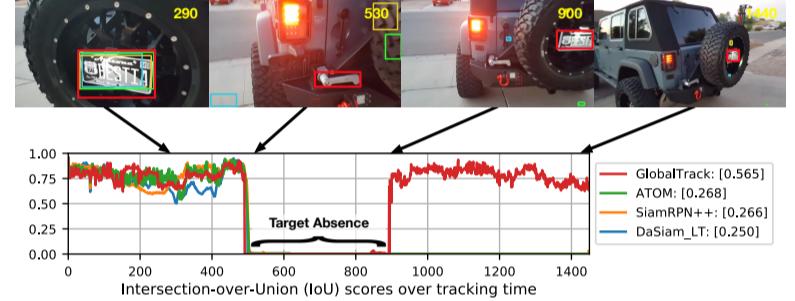


Figure 3: A long-term tracking example where the target undergoes a period of (~ 400 frames) absence. All compared methods fail to relocate the target after the temporary absence, while GlobalTrack relocates it immediately as it reappears, and achieves a much higher average IoU score. The example shows the advantages of GlobalTrack – since it makes no locality assumption and searches globally, its performance in a frame is not affected by its previous failures.

on these datasets, compared to state-of-the-art approaches that typically require complex post-processing. For example, GlobalTrack outperforms SiamRPN++ (Li et al. 2019) and ATOM (Danelljan et al. 2019) by achieving an AUC (area-under-curve) of 52.1% on LaSOT benchmark; it also achieves an large-margin absolute gain of 11.1% on TLP benchmark, over the previous best long-term tracker SPLT (Yan et al. 2019). More importantly, since our approach imposes no assumption on temporal consistency, its performance in one frame is completely independent of previous tracking failures, enabling our model to track very long videos without suffering from cumulative errors. Figure 3 shows an example where the target disappears temporarily for around 400 frames. The compared approaches fail to relocate the target as it reappears, while GlobalTrack catches the target immediately as it shows up, showing the advantage of our approach in long-term tracking scenarios.

Related Work

Long-term tracking refers to the task of continuously locating an arbitrary target in a relatively long video, where the target may temporarily disappear (Valmadre et al. 2018). A key challenge of long-term tracking is to retrieve the target after a period of target absence or tracking failures. To our knowledge, only limited works focus on this task (Kalal, Mikolajczyk, and Matas 2012; Ma et al. 2015; Wang et al. 2019b; Yang et al. 2019). Representative approaches include TLD (Kalal, Mikolajczyk, and Matas 2012), SPL (Supancic and Ramanan 2013), LCT (Ma et al. 2015), EBT (Zhu, Porikli, and Li 2016) and DaSiam_LT (Zhu et al. 2018). TLD, SPL, LCT and EBT are capable of performing full-image search, which is an important ability, especially when the target can disappear. TLD, SPL and LCT searches globally for the target when they detect tracking failures, while EBT always perform full-image search. DaSiam_LT (Zhu et al. 2018) is a variant of DaSiamRPN (Zhu et al. 2018) for long-term tracking, which expands the search area as it detects tracking failures. Similar to these trackers, our approach GlobalTrack can perform full-image search for the target. Differently, we use no online learning and impose no constraints on temporal consistency of the target’s location or scale to avoid cumulative errors.

In terms of the tracking framework, our approach share some similarities with SiamRPN (Li et al. 2018) and ATOM (Danelljan et al. 2019), where they all use shared backbones for extracting features of the query and search images, and they use correlation to encode the relationship between them. However, our work has significant differences with these methods: (1) Our approach performs global search for the target in all frames, without relying on any locality assumption, while ATOM and SiamRPN search only locally and they use complex post-processing to impose constraints on the target’s location and scale changes; (2) We reuse the RPN and RCNN heads of Faster-RCNN and apply them on the modulated features to generate proposals and final predictions. However, SiamRPN uses the query to generate classification and localization weights, while ATOM learns a hand-crafted classifier from scratch during tracking.

Our Approach

In this work, we propose GlobalTrack, a pure global instance search based tracker consisting of two components: a Query-Guided RPN (QG-RPN) for adaptively generating proposals that are specific to a certain query, and a Query-Guided RCNN (QG-RCNN) for classifying the proposals and generating final predictions. The overall architecture of GlobalTrack is shown in Figure 2. In the feature modulation parts of QG-RPN and QG-RCNN, we encode the correlation between the query and search image features in the outputs of the backbone and ROI layers, thereby allowing the model to learn the strong dependency between the queries and the expected prediction results.

In the following, we first introduce the QG-RPN and QG-RCNN in details. Then we present the cross-query loss used in our approach. Finally, we introduce the offline training and online tracking processes of our approach.

Query-guided RPN

Region Proposal Networks (RPNs) are widely used in two-stage object detectors to generate class-agnostic object candidates of pre-defined classes and narrow the search scope down. However, these RPNs are general-purposed; while in tracking, we are only interested in candidates of specific targets. We propose a Query-Guided RPN (QG-RPN) to achieve this, where the key idea is to use correlation to encode the query information in backbone features.

Specifically, let $z \in \mathbb{R}^{k \times k \times c}$ denotes the ROI (Region-Of-Interest) features of the query instance, and $x \in \mathbb{R}^{h \times w \times c}$ represents the search image features, where h , w and k represent feature sizes; we aim to obtain an $\hat{x} \in \mathbb{R}^{h \times w \times c}$ that encodes the correlation between z and x :

$$\hat{x} = g_{qg_rpn}(z, x) = f_{out}(f_x(x) \otimes f_z(z)). \quad (1)$$

Here \otimes denotes the convolution operator, $f_z(z)$ converts z to a convolutional kernel that is applied on the projected features $f_x(x)$ to generate the correlation between z and x . f_{out} ensures the output \hat{x} to have the same size as x . For simplicity, in our approach, we define f_z to be a $k \times k$ convolution layer with zero padding that converts z to a 1×1 convolutional kernel, f_x to be a 3×3 convolution layer with 1 pixel padding, while f_{out} to be a 1×1 convolution layer that converts the channel number back to c . We use no normalization and no activation in these projections.

Since \hat{x} retains the size of x , we directly reuse the modules of RPN and perform its subsequent processes (classification, localization, filtering, etc.) to generate the proposals. We use the same losses of RPN for training QG-RPN, where the classification and localization losses L_{cls} and L_{loc} are binary cross-entropy and smooth L1, respectively (Girshick 2015). The total loss of QG-RPN is:

$$L_{qg_rpn}(z, x) = L_{rpn}(\hat{x}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{loc}} \sum_i p_i^* L_{loc}(s_i, s_i^*) \quad (2)$$

where p_i and s_i are the predicted score and location of the i th proposal, while p_i^* and s_i^* are groundtruths. λ is a weight for balancing the classification and localization losses.

Query-guided RCNN

With proposals generated by QG-RPN, in the second stage, we need to refine the predictions of their labels and bounding boxes according to the ROI features. Note such process is query-specific, since apparently different queries correspond to different groundtruths. We propose Query-Guided RCNN (QG-RCNN) for the classification and bounding box refinement of these proposals. Specifically, given the ROI features of the query $z \in \mathbb{R}^{k \times k \times c}$ and the i th proposal $x_i \in \mathbb{R}^{k \times k \times c}$, we perform feature modulation to encode their correlation:

$$\hat{x}_i = g_{qg_rcnn}(z, x_i) = h_{out}(h_x(x_i) \odot h_z(z)), \quad (3)$$

where \odot denotes the Hadamard production, h_x and h_z are feature projections for x_i and z , respectively, while h_{out} generate output features \hat{x} and ensures that it retains the size

of x_i . In our approach, we simply set h_x and h_z to 3×3 convolution layers with 1 pixel padding, and h_{out} to a 1×1 convolution layer with output channel number being c .

After obtaining the modulated features \hat{x}_i , we continue with the traditional RCNN processes and perform classification and localization on the proposals to obtain the final predictions. During training, similar to QG-RPN, we use binary cross-entropy and smooth L1 as the classification and localization losses to optimize our model. The total loss of QG-RCNN is formulated as:

$$L_{qg_rcnn}(z, x) = \frac{1}{N_{prop}} \sum_i L_{rcnn}(\hat{x}_i), \quad (4)$$

where N_{prop} is the proposal number and

$$L_{rcnn}(\hat{x}_i) = L_{cls}(p_i, p_i^*) + \lambda p_i^* L_{loc}(s_i, s_i^*). \quad (5)$$

Here the p_i and s_i are the estimated confidence and location (center and scale offsets), while p_i^* and s_i^* are groundtruths. λ is a weight for balancing different losses.

Cross-query Loss

To improve the discriminative ability of our approach against instance-level distractors, we propose the cross-query loss. The key idea is to enhance the awareness of our model on the relationship between queries and prediction outputs, by searching a same image using different queries and average their prediction losses. Formally speaking, given a pair of images with M co-exist instances $\{1, 2, \dots, M\}$, we can construct M query-search image pairs, and thereby calculating M QG-RPN and QG-RCNN losses. We average the losses over these queries to obtain the final loss on a pair of images:

$$L_{cql} = \frac{1}{M} \sum_{k=1}^M L(z_k, x), \quad (6)$$

where

$$L(z_k, x) = L_{qg_rpn}(z_k, x) + L_{qg_rcnn}(z_k, x). \quad (7)$$

Note the calculation of L_{cql} is very efficient, since the M query-search image pairs share most computation in backbone feature extraction.

Offline Training

In the training phase, we randomly sample frame pairs from training videos to construct training data. For each frame pair I_z and I_x , we find the M instances that co-exist in both frames, then we construct M query-search image pairs accordingly. We run one forward pass of backbone network on I_z and I_x to obtain their feature maps, then we perform M times of feature modulation using Eq. (1) and Eq. (3). The modulated features are fed into QG-RPN and QG-RCNN to obtain the query-specific predictions. The loss of the frame pair is then evaluated using Eq. (6), which is an average of losses over all available queries. The data sampling and loss evaluation are performed for several iterations, and we use the standard stochastic gradient descent (SGD) algorithm to optimize our model.

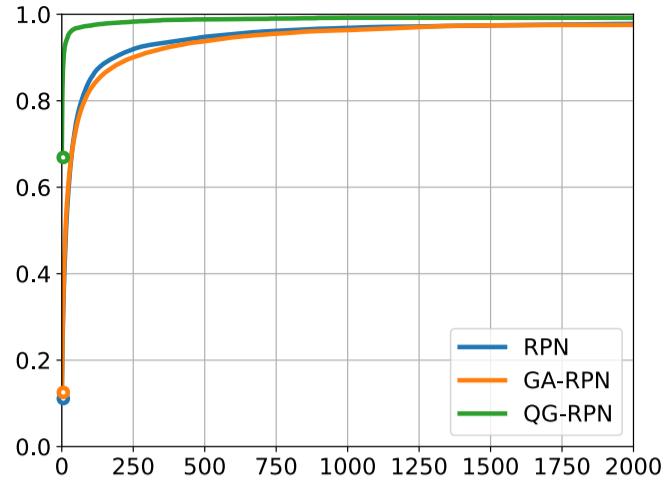


Figure 4: AR@k plots of different region proposal networks (RPNs). Compared to general purposed proposal networks RPN and GA-RPN, the proposed query-guided RPN consistently show a much higher recall.

Table 1: AR@k of different region proposal networks (RPNs). QG-RPN shows high recall even with very few proposals. Moreover, with all proposals available, the average recall of QG-RPN still surpasses the compared approaches.

(%)	AR@1	AR@10	AR@100	AR@500	AR@2000
RPN	11.3	40.5	84.9	94.8	97.8
GA-RPN	12.9	43.3	82.6	93.8	97.5
QG-RPN	67.1	91.9	97.4	98.8	99.2

Online Tracking

The tracking process of our approach is extremely simple. In the first frame, we initialize the query using the user-specified annotation. Then the query is fixed throughout the tracking process with no updating. In a new tracking frame, we run a forward pass of QG-RPN and QG-RCNN with the query and current image as inputs. Finally, we directly take the top-1 prediction of QG-RCNN in this frame as the tracking result. No further post-processing is used. Although adding some post-processing, such as casting penalization on large state variations or performing trajectory refinement, may improve the performance of our approach, we prefer to keep the current model simple and straightforward, and leave a more adaptive tracking model to our future works.

Experiments

To verify the effectiveness of our approach, we conduct evaluation on four large-scale tracking benchmarks: LaSOT (Fan et al. 2019), TrackingNet (Muller et al. 2018), TLP (Moudgil and Gandhi 2018) and OxUvA (Valmadre et al. 2018), where their test sets contain 280, 511, 180 and 166 videos, respectively. LaSOT, TLP and OxUvA are long-term tracking benchmarks with average video lengths of 2500, 13529 and 4260 frames, respectively. We compare our overall results with state-of-the-art approaches in this section. We also carry out experiments to analyze the effectiveness of our individual components.

Table 2: Numerical comparison of QG-RPN and QG-RCNN in AR@k. QG-RCNN outperforms QG-RPN in AR@1 by around 9.5%. As more candidates are introduced, QG-RPN achieves higher average recall than QG-RCNN by around 5%. As the number of candidates gets larger, the average recall of QG-RPN and QG-RCNN become close.

(%)	AR@1	AR@5	AR@10	AR@100	AR@512
QG-RPN	67.1	87.1	91.9	97.4	98.8
QG-RCNN	76.6	82.5	85.9	97.4	98.9

Table 3: Comparison of GlobalTrack trained using cross-query loss and single-query loss. The performance is evaluated on the test set of LaSOT. The results show that the model trained with cross-query loss consistently outperforms that trained with single-query loss in all three metrics.

(%)	Precision	Norm. Precision	Success
Single-query loss	49.3	55.7	49.5
Cross-query loss	52.7	59.9	52.1

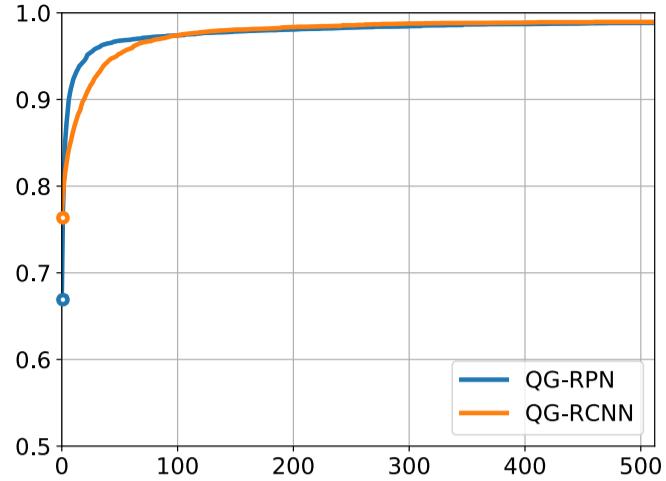


Figure 5: AR@k plots of QG-RPN and QG-RCNN. QG-RCNN achieves much higher AR@1 (equivalent to the top-1 accuracy) than QG-RPN, while its average recall is soon surpassed by QG-RPN as more candidates are introduced. The results demonstrate the different preferences of QG-RPN and QG-RCNN in recall and accuracy.

Implementation Details

Parameters We use Faster-RCNN with ResNet-50 backbone (Girshick 2015) as our base model for constructing query-guided RCNN. The channel number of the backbone features is $c = 256$. We set the output channel number of f_x, f_z and h_x, h_z to $c' = 256$ as well. In this way, the f_z in effect converts the query features z to a 1×1 depth-wise convolutional kernel, which is more efficient in computation. We employ ROI align (He et al. 2017) to extract ROI features, where the output feature size is set to $k = 7$. We normalize the size of each input image so that its longer edge is no larger than 1333 pixels, while its shorter edge is no larger than 800 pixels. The QG-RPN generates 2000 proposals for calculating its losses, of which 512 proposals are sampled to be fed into QG-RCNN. The localization loss weight in Eq. (2) and Eq. (5) is set to $\lambda = 1$.

Training Data We use a combination of COCO (Lin et al. 2014), GOT-10k (Huang, Zhao, and Huang 2018) and LaSOT (Fan et al. 2019) datasets for training our model, where the sampling probabilities of the three datasets are 0.4, 0.4 and 0.2, respectively. COCO is an image object detection dataset containing over 118 thousand images, which belong to 80 object classes. GOT-10k and LaSOT are visual tracking datasets, where GOT-10k consists of 10,000 videos belonging to 563 object classes, while LaSOT consists of 1,400 videos belonging to 70 object classes. For COCO dataset, we randomly sample an image and perform data augmentation on it to generate an image pair; while for GOT-10k and LaSOT datasets, we directly sample frame pairs from videos. We use random horizontal flipping and color jitter to augment the image pairs and to enrich our training data.

Optimization We use stochastic gradient descent with a batch size of 4 pairs to train our model. The momentum and weight decay are set to 0.9 and 1×10^{-4} , respectively. The backbone of our model is initialized from Faster-RCNN pre-

tained on COCO dataset. We fix the batch normalization parameters during training while allowing all other parameters trainable. We train our model for 12 epochs on COCO and another 12 epochs on a combination of COCO, GOT-10k and LaSOT datasets, as described in the previous subsection. The initial learning rate is set to 0.01, and it decays with a factor of 0.1 at epoch 8 and 11. Our approach is implemented in Python, using PyTorch. The training process takes about 16 hours on four GTX TitanX GPUs, while the online tracking runs at around 6 fps on a single gpu.

Ablation Study

In this section, we perform an extensive analysis of the proposed approach. Unless specified, the experiments are conducted on OTB-2015 dataset, which consists of 100 videos. We evaluate the approaches using overlap precision (OP), which indicates the percentage of successfully tracked frames where the overlap rates exceed 0.5.

Analysis of QG-RPN We compare the proposed query-guided RPN, which is able to adaptively change the predictions according to different queries, with the general purposed RPN (Girshick 2015) and its improved variant Guided-Anchor RPN (GA-RPN) (Wang et al. 2019a). The results are shown in Figure 4 and Table 1. We use AR@k to evaluate these RPNs, which indicates the average recall using top-k predictions. As shown in Figure 4, QG-RPN consistently outperforms RPN and GA-RPN by a large margin, and it is able to achieve high recall with only a few proposals. Table 1 shows the numerical comparison. Using only the top-1 prediction, QG-RPN achieves a recall of 67.1%, surpassing RPN and GA-RPN by over 55%. With only the top-10 proposals, QG-RPN achieves a recall of 91.9%, which is comparable with the top-500 recalls of RPN and GA-RPN, indicating the efficiency of our approach. With top-2000 proposals, our approach also outperforms the compared approaches by achieving a recall of 99.2%. These results verify

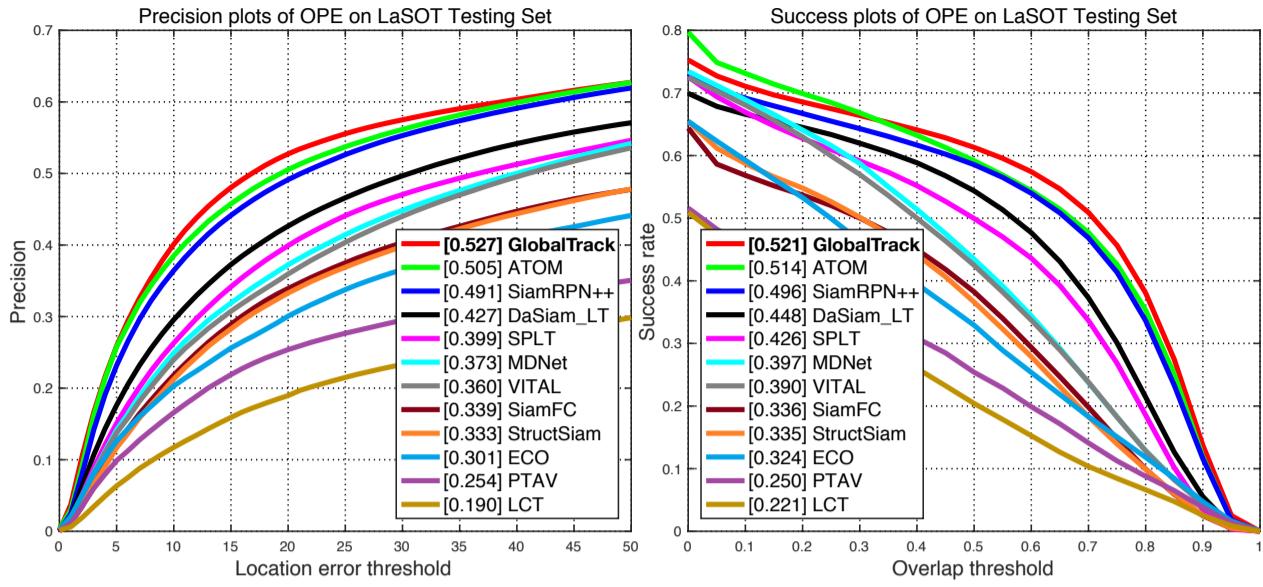


Figure 6: Success plots of GlobalTrack and state-of-the-art trackers on the test set of LaSOT. Compared to the previous best tracker ATOM, our approach achieves absolute gains of 2.2% in precision score and 0.7% in success score, respectively. Besides, GlobalTrack also outperforms the latest long-term trackers SPLT and DaSiam_LT by a large margin.

Table 4: State-of-the-art comparison on the long-term tracking benchmark TLP in terms of success rate (SR, under overlap threshold 0.5), success score and precision score.

(%)	SR _{0.5}	Success	Precision
LCT	8.7	9.9	7.2
ECO	21.9	20.2	21.2
ADNet	22.1	22.3	20.3
MDNet	42.3	37.0	38.4
MBMD	48.1	-	-
SiamRPN	51.5	-	-
ATOM	47.5	-	-
SPLT	52.7	-	-
GlobalTrack	63.8	52.0	55.6

Table 5: State-of-the-art comparison on the test set of TrackingNet in terms of precision, normalized precision and success (AUC).

(%)	Precision	Norm. Precision	Success
CFNet	53.3	65.4	57.8
MDNet	56.5	70.5	60.6
ECO	49.2	61.8	55.4
UPDT	55.7	70.2	61.1
DaSiamRPN	59.1	73.3	63.8
SiamRPN++	69.4	80.0	73.3
C-RPN	61.9	74.6	66.9
ATOM	64.8	77.1	70.3
GlobalTrack	65.6	75.4	70.4

the effectiveness of QG-RPN and demonstrate its superiority in both recall rate and efficiency.

Comparison of QG-RPN and QG-RCNN QG-RPN and QG-RCNN are the two stages of our approach that pursue for recall and the top-1 accuracy, respectively. We compare their AR@k results in Figure 5 and Table 2. As shown in Figure 5, the QG-RCNN achieves higher AR score than QG-RPN at the very start, and is soon surpassed by QG-RPN as more candidates are available. From Table 2 we observe that, the top-1 accuracy (which is exactly the AR@1) of QG-RCNN surpasses QG-RPN by up to 9.5%. However, as more candidates are introduced, QG-RPN achieves significantly higher recall than QG-RCNN, even using only the top-5 predictions. As k gets even larger (e.g., $k \geq 100$), the ARs of QG-RPN and QG-RCNN become close. The results verify the effectiveness of QG-RPN and QG-RCNN, and show their different preferences for accuracy and recall.

Impact of Loss Function To verify the effectiveness of the proposed cross-query loss, we compare our model trained with cross-query loss and with single query loss – the loss that only considers one instance per image pair. We

evaluate the two models on the test set of LaSOT dataset. The results are shown in Table 3. We observe that, the model trained with cross-query loss surpasses that trained with the single-query loss in all three metrics by 2.6% ~ 4.2%, verifying the robustness of the proposed loss function.

Comparison with State-of-the-art

We compare our approach GlobalTrack with state-of-the-art trackers on four large-scale tracking benchmarks. The compared approaches include SPLT (Yan et al. 2019), ATOM (Danelljan et al. 2019), SiamRPN++ (Li et al. 2019), C-RPN (Fan and Ling 2019), MBMD (Zhang et al. 2018), DaSiam_LT (Zhu et al. 2018), UPDT (Bhat et al. 2018), VITAL (Song et al. 2018), SINT (Tao, Gavves, and Smeulders 2016), CF2 (Chao et al. 2016), ADNet (Yun et al. 2017), MDNet (Nam and Han 2016), SiamFC (Bertinetto et al. 2016), SiamFCv2 (Valmadre et al. 2017), CFNet (Valmadre et al. 2017), StructSiam (Yunhua et al. 2018), ECO (Danelljan et al. 2017), PTAV (Fan and Ling 2017), TLD (Kalal, Mikolajczyk, and Matas 2012) and LCT (Ma et al. 2015).

Table 6: State-of-the-art comparison on the **test set** of OxUvA in terms of maximum geometric mean (MaxGM), true positive rate (TPR) and true negative rate (TNR).

(%)	MaxGM	TPR	TNR
ECO-HC	31.4	39.5	0.0
MDNet	34.3	47.2	0.0
TLD	43.1	20.8	89.5
LCT	39.6	29.2	53.7
SINT	32.6	42.6	0.0
SiamFC+R	45.4	42.7	48.1
EBT	28.3	32.1	0.0
DaSiam_LT	41.5	68.9	0.0
MBMD	54.4	60.9	48.5
SPLT	62.2	49.8	77.6
GlobalTrack	60.3	57.4	63.3

Table 7: State-of-the-art comparison on the **development set** of OxUvA in terms of maximum geometric mean (MaxGM), true positive rate (TPR) and true negative rate (TNR).

(%)	MaxGM	TPR	TNR
ECO-HC	26.6	28.3	0.0
MDNet	32.4	42.1	0.0
TLD	36.6	14.1	94.9
LCT	31.7	22.7	43.2
SINT	30.3	26.8	0.0
SiamFC+R	39.7	35.4	43.8
GlobalTrack	63.9	55.8	73.2

LaSOT The test set of LaSOT consists of 280 videos with an average length of 2448 frames, which is longer than most other datasets. The precision and success plots on LaSOT are shown in Figure 6. Compared to the previous best approaches ATOM and SiamRPN++, our approach achieves an absolute gain of 2.2% and 3.6% in precision score, and 0.7% and 2.5% in success score, respectively. The tracking results demonstrate the advantage of global instance search in long-term tracking.

TLP The dataset consists of 50 long videos with an average length of 13529 frames, which is much longer than all other tracking datasets. The results of the trackers are shown in Table 4. Our approach GlobalTrack outperforms all other trackers, including the latest long-term tracker SPLT, by a very large margin ($\sim 11.1\%$ absolute gain in SR_{0.5}), verifying the significant advantage of our approach over all existing algorithms in tracking very long videos.

TrackingNet The test set of TrackingNet consists of 511 videos collected from the YouTube website. The evaluation results of the trackers are shown in Table 5. SiamRPN++ and ATOM achieve impressive success scores of 73.3% and 70.3%, respectively, while our approach achieves a success score of 70.4%, which is comparable with the best trackers. The tracking results show the generalization ability of our approach on large test data.

OxUvA OxUvA is a long-term tracking benchmark, where its development and test sets consist of 200 and 166 videos, respectively. The average video length of OxUvA is 4260 frames, which is much longer than most other datasets. The protocol of OxUvA requires the approach to submit not only the estimated bounding boxes, but also the prediction of target absences in all frames. To generate the presence/absence prediction, we simply threshold the top-1 scores of QG-RCNN, where the frames with top-1 scores exceed $\tau = 0.84$ are considered *target presence*, while those with top-1 scores below the threshold are considered *target absence*. The evaluation results of the trackers on the **test and development sets** of OxUvA are shown in Table 6 and Table 7, respectively. Compared to the reported best tracker SiamFC-R, our approach achieves an absolute gain of 14.9%

and 24.2% on the test and development sets of OxUvA in terms of MaxGM (maximum geometric mean of TPR and TNR), respectively. The results demonstrate the strong performance of our approach in long-term tracking scenarios.

Conclusions and Future Works

In this work, we propose a pure global instance search based tracker that imposes no assumption or constraint on temporal consistency. Therefore, its performance in a frame is not affected by previous tracking failures, which makes it ideal for long-term tracking. The method is developed based on two-stage object detectors, and it is composed of two components: a query-guided region proposal network for generating query-specific instance candidates, and a query-guided region convolutional neural network for classifying these candidates and generating the final predictions. Experiments on four large-scale tracking benchmarks verify the strong performance of the proposed approach.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China (Grant No. 2016YFB1001005), the National Natural Science Foundation of China (Grant No. 61602485 and No. 61673375), and the Projects of Chinese Academy of Science (Grant No. QYZDB-SSW-JSC006).

References

- Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, 850–865. Springer.
- Bhat, G.; Johnander, J.; Danelljan, M.; Khan, F. S.; and Felsberg, M. 2018. Unveiling the power of deep tracking.
- Chao, M.; Huang, J. B.; Yang, X.; and Yang, M. H. 2016. Hierarchical convolutional features for visual tracking. In *IEEE International Conference on Computer Vision*.
- Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; and Felsberg, M. 2017. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6638–6646.
- Danelljan, M.; Bhat, G.; Khan, F. S.; and Felsberg, M. 2019. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6638–6646.

- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*
- Fan, H., and Ling, H. 2017. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking.
- Fan, H., and Ling, H. 2019. Siamese cascaded region proposal networks for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7952–7961.
- Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; and Ling, H. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 2961–2969.
- Huang, L.; Zhao, X.; and Huang, K. 2018. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*.
- Huang, L.; Zhao, X.; and Huang, K. 2019. Bridging the gap between detection and tracking: A unified approach. In *Proceedings of the IEEE International Conference on Computer Vision*, 3999–4009.
- Kalal, Z.; Mikolajczyk, K.; and Matas, J. 2012. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence* 34(7):1409–1422.
- Li, B.; Yan, J.; Wu, W.; Zhu, Z.; and Hu, X. 2018. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8971–8980.
- Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; and Yan, J. 2019. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.
- Ma, C.; Yang, X.; Zhang, C.; and Yang, M. H. 2015. Long-term correlation tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Moudgil, A., and Gandhi, V. 2018. Long-term visual object tracking benchmark. In *Asian Conference on Computer Vision*, 629–645. Springer.
- Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; and Ghanem, B. 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision*, 300–317.
- Nam, H., and Han, B. 2016. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4293–4302.
- Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.; and Yang, M. H. 2018. Vital: Visual tracking via adversarial learning.
- Supancic, J. S., and Ramanan, D. 2013. Self-paced learning for long-term tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2379–2386.
- Tao, R.; Gavves, E.; and Smeulders, A. W. M. 2016. Siamese instance search for tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; and Torr, P. H. 2017. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2805–2813.
- Valmadre, J.; Bertinetto, L.; Henriques, J. F.; Tao, R.; Vedaldi, A.; Smeulders, A. W.; Torr, P. H.; and Gavves, E. 2018. Long-term tracking in the wild: A benchmark. In *Proceedings of the European Conference on Computer Vision*, 670–685.
- Wang, J.; Chen, K.; Yang, S.; Loy, C. C.; and Lin, D. 2019a. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2965–2974.
- Wang, X.; Sun, T.; Yang, R.; and Luo, B. 2019b. Learning target-aware attention for robust tracking with conditional adversarial network.
- Wu, Y.; Lim, J.; and Yang, M.-H. 2015. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9):1834–1848.
- Yan, B.; Zhao, H.; Wang, D.; Lu, H.; and Yang, X. 2019. 'skimming-perusal' tracking: A framework for real-time and robust long-term tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, 2385–2393.
- Yang, R.; Zhu, Y.; Wang, X.; Li, C.; and Tang, J. 2019. Learning target-oriented dual attention for robust rgb-t tracking. In *2019 IEEE International Conference on Image Processing*, 3975–3979. IEEE.
- Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; and Young Choi, J. 2017. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2711–2720.
- Yunhua, Z.; Lijun, W.; Dong, W.; Mengyang, F.; Huchuan, L.; and Jinqing, Q. 2018. Structured siamese network for real-time visual tracking.
- Zhang, Y.; Wang, D.; Wang, L.; Qi, J.; and Lu, H. 2018. Learning regression and verification networks for long-term visual tracking. *arXiv preprint arXiv:1809.04320*.
- Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; and Hu, W. 2018. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision*, 101–117.
- Zhu, G.; Porikli, F.; and Li, H. 2016. Beyond local search: Tracking objects everywhere with instance-specific proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 943–951.