TLPG-Tracker: Joint Learning of Target Localization and Proposal Generation for Visual Tracking

Siyuan Li[1], **Zhi Zhang**[2*], **Ziyu Liu**[3], **Anna Wang**[4], **Linglong Qiu**[2] and **Feng Du**[3]
[1]Nanjing University, Nanjing, China
[2]Zhijiang College of Zhejiang University of Technology, Shaoxing, Zhejiang, China
[3]Zhejiang University of Technology, Hangzhou, China
[4]Sichuan University, Chengdu, China
lsy@smail.nju.edu.cn, zhang2@kth.se, {lzyu, fun}@zjut.edu.cn, {annascu98, qll7521}@gmail.com

Abstract

Target localization and proposal generation are two essential subtasks in generic visual tracking, and it is a challenge to address both the two efficiently. In this paper, we propose an efficient two-stage architecture which makes full use of the complementarity of two subtasks to achieve robust localization and high-quality proposals generation of the target jointly. Specifically, our model performs a novel deformable central correlation operation by an online learning model in both two stages to locate new target centers while generating target proposals in the vicinity of these centers. The proposals are refined in the refinement stage to further improve accuracy and robustness. Moreover, the model benefits from multi-level features aggregation in a neck module and a feature enhancement module. We conduct extensive ablation studies to demonstrate the effectiveness of our proposed methods. Our tracker runs at over 30 FPS and sets a new state-of-the-art on five tracking benchmarks, including LaSOT, VOT2018, TrackingNet, GOT10k, OTB2015.

1 Introduction

Visual object tracking is a fundamental problem in computer vision which is widely applied in automatic driving and video surveillance, etc. Given an initial bounding box of an arbitrary target in the first frame, it aims to track the target automatically in every frame that follows. The tracking problem is usually decomposed into a classification task for the target localization and a regression task for the target state estimation. Commonly, the target location and state are represented as the target center and a bounding box.

Mainstream trackers perform tracking via a template-matching strategy which is well-performed in both speed and accuracy. Following the modern trend in object detection, we integrate a general pipeline for template-matching trackers in Figure 1, which uses two siamesed branches and two separate heads for the classification and the box estimation in a tracking problem. The template-matching methods store the target

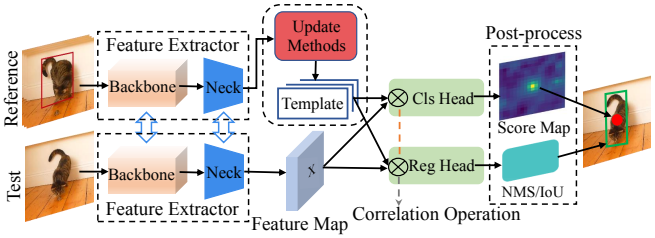

Figure 1: An overview of a general template-matching pipeline in a Siamese-like way. Sharing the same backbone and neck as the feature extractor, a test branch is used for tracking the test frame while a reference branch stores template from reference frames. Some update methods are employed in online training trackers. The classification (localization) and the proposal boxes regression (estimation) are performed separately in subnet heads. The correlation operation between the feature map x and the target model is the core operation in these trackers.

appearance information in a template formulate the tracking problem by learning to generate a score map by correlation operations between features representations of the template and the search region. However, most trackers overlook the inherent relation between target localization with discriminative features and target shape estimation, and try to optimize two subtasks separately. It results in some inevitable shortfalls, e.g., the lack of discriminative abilities in Siamese-like trackers [Li *et al.*, 2018a] and the lack of box estimation capacities in Discriminative Correlation Filter (DCF) methods [Danelljan *et al.*, 2017].

To overcome aforementioned limitations, we propose a novel architecture to learn target localization and proposal generation jointly in an online learning manner. Inspired by the recently proposed RepPoints [Yang *et al.*, 2019], we introduce a deformable central correlation operation, referred as DCC, to exploit the relation between discriminative features for target-background classification and spatial detailed features for target state estimation. And we propose a two-stage process in which we further refine the proposal and improve robustness. To meet the requirements of the model for both fine-grained target representations and sematic features, we design a balanced neck and an effective feature enhancement module to aggregate multi-level features. Our final tracking architecture is a unified multi-task network with end-to-end learning methods. The main contributions of this paper are

*Zhi Zhang is the corresponding author.

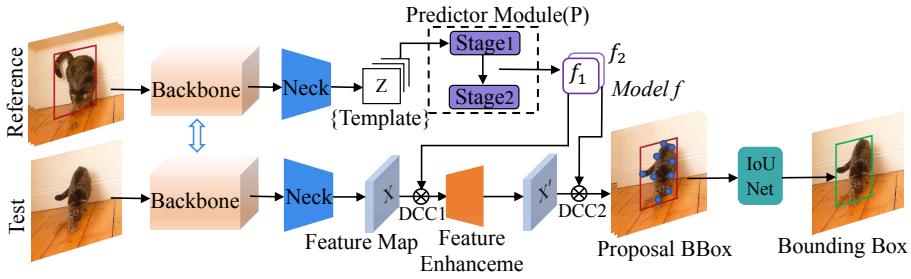

Figure 2: An overview of our tracking architecture. During inference, a test frame is input to the test branch (bottom left), and proposal bounding boxes are the outputs of a two-stage generation and refinement process with DCC1 and DCC2 which refer to deformable central correlation in stage 1 and 2. The final output is the proposal with the highest IoU score or predicted score.

three-fold:

- Online learning models with novel deformable central correlations are developed to perform robust target localization and accurate proposal generation jointly;
- An efficient two-stage architecture is designed for more precise box estimation;
- A balanced neck and a feature enhancement module are proposed to extract aggregated features for two-stage tracking.

With the above contributions, our proposed tracker, referred as TLPG-Tracker, establishes a new state-of-the-art in comprehensive experiments while possessing in real-time at over 30 FPS on five challenging benchmarks: LaSOT [Fan *et al.*, 2019], GOT10k [Huang *et al.*, 2018], VOT2018 [Kristan *et al.*, 2018], TrackingNet [Muller *et al.*, 2018], OTB-100 [Wu *et al.*, 2015]. We further provide an extensive experiment to analyze the impact of proposed component with other existing methods.

2 Related Work

Most of the modern trackers are performed in a template-matching strategy including DCF approaches [Danelljan *et al.*, 2017; Bhat *et al.*, 2018] and Siamese networks [Bertinetto *et al.*, 2016; Li *et al.*, 2018a]. An effective correlation operation is introduced into visual tracking by seminal work MOOSE [Bolme *et al.*, 2010] to match an explicit template with a search patch with learnable weights in the frequency domain. After the rise of deep learning, DCF-based methods with deep sematic features can achieve impressive robustness in target localization. However, they are hard to make full use of the fine-grained features to estimate target shapes.

In contrast to DCF, the recent Siamese networks [Bertinetto *et al.*, 2016; Li *et al.*, 2018a], which computes cross-correlation similarities between a template and a search region in an embedding space, achieve high accuracy and fast tracking speed. Many Siamese-based trackers can balance target classification and state estimation tasks with their extensive architectures in Figure 1. SiamRPN [Li *et al.*, 2018a] imports an anchor-based region proposal networks from object detection tasks, and its extensions [Zhu *et al.*, 2018; Li *et al.*, 2019] show efficiency of bounding box regression with better discriminative abilities. To gain more accurate box estimation, ATOM [Danelljan *et al.*, 2019] employs an IoU-Net [Jiang *et al.*, 2018] based module to predict Intersection-over-Union (IoU) overlap and select best proposal boxes. More recently, an online learning discriminant models with Siamese-based structure are proposed [Bhat *et al.*, 2019] to make up for the deficiency of discriminative abilities in Siamese-based methods. Overall, they still optimize the classification and estimation separately.

3 Method

In this work, we develop a novel joint learning architecture to exploit the relation between target localization and proposal generation with two main principles: (1) the most discriminative features of a target usually locate in the vicinity of the target center while the target bounding box is supported by fine-granted features near target borders to fit its shapes; (2) the target-background classification task and the proposal generation and regression task are complementary in tracking. In terms of the general pipeline, we split our Siamese-like network into a feature exactor with a neck module, online learning models which perform target localization and proposal generation by deformable central correlation and a refinement stage with a feature enhancement module, shown in Figure 2.

In our two branches framework, feature maps for the test and the reference branches are extracted from backbone network with a balanced neck module which fuses the sematic and spatial information from multi-layer features. The target models are constituted by deformable convolution kernels which are learned online on the target template Z (reference) during inference time and perform two-stage tracking. In stage 1, the target in a test frame is tracked by deformable central correlation between the model f and a feature map x of the search region (test) which finds coordinates where suitable target proposal boxes are formed. Followed by a feature enhance module which aggregates multi-level features, the proposal boxes are refined by another deformable central correlation on the enhanced feature x' in stage 2. We get final output bounding boxes with optional post-process, e.g., non-maximum suppression as NMS and IoU-Net. We update weights of the target models by an online-training optimizer module P.

3.1 Deformable Convolution in Visual Tracking

In order to facilitate the introduction of our target models, we briefly revisit deformable convolution [Dai *et al.*, 2017] in the context of visual tracking. For notation clarity, we describe

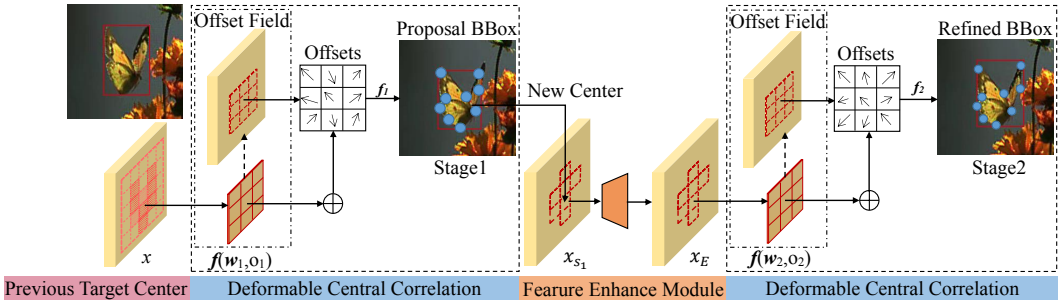

Figure 3: An overview of the proposal generation and refinement module. With a feature map x, discriminative features localization and proposal generation are performed by deformable central correlation near the previous target center in stage 1. Refinement are performed on top N_c proposals center points in stage 1 after the feature enhancement to get refined bounding boxes in stage 2.

the convolution modules in 2D spatial domain, and the operation remains the same in each channel in 3D case.

Deformable convolution can be regarded as a self-attention instantiations of spatial attention mechanisms [Zhu *et al.*, 2019]. From the aspect of query-key pairs in spatial attention, an element on each coordinate of a feature map is regarded as a query element while the elements on sampled coordinates are regarded as key elements. Based on regular convolution, which has fixed sampling grid region R and convolutional weight w, the deformable convolution models the relation between a query with its corresponding keys by augmenting learnable 2D offsets $\{\Delta p_k\}_{k=1}^{N_s}$ to R on integral spatial locations in a feature map $x \in \mathbb{R}^{W\times H}$ which are stored in offset field $o \in \mathbb{R}^{W\times H\times 2N_S}$. N_s refers to the number of sample points in a deformable convolution kernel. For each coordinate p_i on output feature map $y \in \mathbb{R}^{W\times H}$ we have output as

$$y(p_i) = \sum_{p_r\in R} w(p_r) * x(p_i + p_r + \Delta p_k) \tag{1}$$

In practice, a bilinear interpolation kernel $A(.,.)$ is used to map the fractional $p = p_i+p_r+\Delta p_k$ to an integral coordinate.

In generic tracking, it is a common sense that coarse localization of the target center is much easier than finding an accurate target bounding box. The main reason is that features near the border of the target which determine the shape of the bounding box are unstable and usually mixed with background, while features near the target center are more constant and fixed to some extent. Intuitively, we can learn the relation between discriminative features near target centers and features for target shapes by deformable convolution to generate proper bounding boxes. Inspired by RepPoints, we collect the sampled points of deformable convolution at the coordinate p_0 in a feature map as

$$P_{key} = \{p_u\}_{u=1}^{N_S} \tag{2}$$

where $p_u \in \mathbb{R}^2$ refers to a position on feature map x. We define a converting function $\Gamma : P_{key} \rightarrow P$, to transform this point-based representation to a proposal bounding box. Specially, the proposal P is formed by the distances from the coordinate p_0 to the left-top and bottom-right points as 4D real vector (l, t, r, b). We adopt a min-max function, which performs min-max operation over two axes of P_{key} to determine left-top and bottom-right points of P, as the function Γ.

3.2 Deformable Central Correlation

In this section, we detail how to learn target models and perform an online learning deformable central correlation for tracking. Generally, given a set of training sample $S_{train} = \{(x_i, B_i)\}_{i=1}^{n}$, where x_i is a feature map from the feature extractor and $B_i = (l_i, t_i, r_i, b_i)$ is the 4D corresponding bounding box, the target model f is learned online or offline. In recent template-matching trackers, there are a classification task and an estimation task. As for the classification, it usually performs correlation between the template and the input feature map x_i by the model f to discriminate the explicit target from background distractors, and outputs a score map $s = x_i * f$ where $s \in \mathbb{R}^{W\times H}$ and $*$ denotes correlation operation. The online learning classification loss is formulated as

$$l_{cls} = \frac{1}{|S_{train}|}\sum_{i=1}^{n}\|d(s, y_i)\|^2 + \|\lambda f\|^2 \tag{3}$$

where the function $d(s, y_i)$ computes the residual at every spatial location on the predicted score map s and a corresponding label y_i, and λ is a regularization factor. As for the estimation, the traditional methods generate N_p proposal bounding boxes $\{P_{it}\}_{t=1}^{N_p}$ by anchor strategies SiamRPN. The regression loss is formulated as

$$l_{reg} = \frac{1}{|S_{train}|}\sum_{i=1}^{n} r\left(p_{it}, B_i\right) \tag{4}$$

where the function $r\left(P_{it}, B_i\right)$ computes the residual between the proposal and a bounding box. As mentioned in 3.1, we adopt a deformable convolution layer in the online learning model f which is defined as $f(x; w, o)$ and aim to find a coordinate p^* which generates the most precise proposal bounding box P^*. The deformable central correlation shown in Figure 3 is carefully designed with an observation that the most representative features are always near the center of mass of the target rather than the bounding box center.

Proposal Generation. Given a location p_j on the feature map x_i, our model will generate a proposal bounding box P_j with sampled points. We expect to generate accurate proposal bounding boxes of the target on discriminative features around the vicinity of the target center. So, we use two IoU thresholds θ_{hi} and θ_{lo} on the classification label y_i to separate positive and negative training samples. For a positive sample $p_j = (p_x, p_y)$ which has $\mathrm{IoU}(P_j, B_j) \geq \theta_{hi}$, the regression target P_j^* is formulated as

$$l^* = p_x - l_i, t^* = p_y - t_i, r^* = r_i - p_x, b^* = b_i - p_y \tag{5}$$

We adopt the smooth l_1 distance to the function $r(P_j, P_j^*) = Smooth_{l1}(P_j, P_j^*)$ which is easy to optimize online. For a negative sample p_j which has $y_i(p_j) < \theta_{lo}$, we simply minimize the IoU score which is non-negative between P_j and B_j using the IoU loss as the function $r(P_j, B_j) = \text{IoU}(P_j, B_j)$ in offline training.

Target Localization. To ensure the discriminative capacity, the model f is commonly defined as a discriminative filter to locate the target center (c_x, c_y). However, this definition is usually faces two problems: (1) data imbalance which makes the model mainly focus on easy negatives; (2) inherent ambiguity of the classification label y_i. Benefitted from our proposal generation, the center of mass of the target can be roughly clustered. Thus, we define $CR_i(c_x, c_y, \sigma_c w_b, \sigma_c h_b)$ as the center region of the bounding box B_i, which roughly covers potential target centers, with a shrink scale $\sigma_c \in (0, 1)$. Note that the label of each spatial coordinate inside CR_i is set to 1 while outside the bounding box B_i is set 0. We ignore the region between CR_i and B_i. The intermediate coordinates are smoothed with Gaussian distribution centered at the box center. We formulate the function $d(s, y_i)$ in a hinge-like form [Bhat *et al.*, 2019],

$$d(s, y_i) = (y_i s + (1 - y_i)\max(0, s) - y_i) \tag{6}$$

Both the loss (3) and (4) are applied in online and offline training. The predicted scores of the proposal which reflect both localization and estimation are more suitable for the NMS procedure.

3.3 Center-bias Refinement with Feature Enhancement

To get more accurate results, we introduce a two-stage refinement method with a multi-layer feature enhancement module. Note that the ResNet architecture is employed as backbone and the feature map x has a spatial stride of 16.

Center-bias Refinement. Our ablation experiments show that multi-stage bounding box regression, which learns the bias between the proposals and the ground truth, yields poor performance (see section 4.1). Unlike the regression methods with fixed centers in anchor-based strategies, the proposed method gains refinement with the bias of the target center locations. As shown in Figure 3, we pay more attention to search representative target features and the target center in the region of interest in stage 1, and focus on target aligning in stage 2 with fewer proposals. We select top N_c centers of the proposal bounding boxes in stage 1 by their IoU scores. After removing the duplicate locations, they are used as new centers to generate new proposals in stage 2.

Feature Enhancement. As we discussed in section3.2, the deformable central correlation actually requires fine-grained and sematic features. Although the feature map x_{S1} gains more target-related features after stage 1, it losses some fine-grained sematic features around the target because of the discriminant ability of the model. Thus, we use a multi-resolution aggregation method to fuse fine-grained features from multi-level feature maps including C_3, C_4, C_5 with spatial stride of 8, 16, 32, shown in Figure 4. We import refined

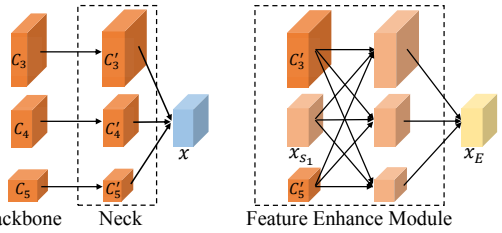

Figure 4: Architecture of the balanced neck and the feature enhancement module. We aggregate multi-level feature maps with the balanced neck by rescaling and explore sematic features and spatial details for the refinement stage in a full-connected manner.

features form C_3' and C_5' directly from the neck (see section3.4) like residual blocks to form three parallel branches. Then we fuse three parallel branches in a fully-connection manner and combine them to get the enhanced feature map x_E. Thus, we use a multi-resolution aggregation method to fuse fine-grained features from multi-level feature maps, as shown in Figure 4. Since more spatial details of target appearance are in lower convolution layers, we import high-resolution feature map x_{F1} directly from backbone like a residual block to keep spatial features. We form three parallel branches to get fully aggregated features. To keep the rich sematic features of the target in current feature map x_{S1}, we perform regular convolution on x_{S1} to get $x_{S1}{}'$ with low resolutions. We use 1×1 convolution to unified the number of channels in x_{F1} and $x_{S1}{}'$. Then we fuse three parallel branches in a fully-connection manner and combine them to get the enhanced feature map x_E.

3.4 Balanced Neck

The proposed balanced neck module serves as a connection between a backbone and head networks, and outputs a single feature map which balances spatial details and robust sematic features for the head network.

We observe that different feature levels at different resolutions of a backbone usually contribute unequally to the fused output. Thus, we design a simple neck structure with bounded weight n_i (per-feature), where $\Sigma_i n_i = 1$. As for the ResNet architecture, we first unify the multi-level features C_3, C_4, C_5 to the same channel number as C_4 with 1×1 convolution as C_3', C_4', C_5'. Unlike the bottom-top integrating strategy used in FPN [Lin *et al.*, 2017], we balance semantic and fine-grained features by rescaling and averaging them with a scalar weight as

$$x = \frac{1}{3}\sum_{l=3}^{5} n_l C_l' \tag{7}$$

Note that C_3 and C_5 are resized to C_4 as C_3' and C_5' by average-pooling and linear interpolation after 1×1 convolution respectively.

3.5 Implementation Details

As for online learning, we set the kernel size of DCC to 5×5 in our target models for two stages. To speed up convergence of online training, we use Steepest Descent (SD) in DiMP. We train 10 iterations at the first frame as initialization. The models are updated in an optimizer module P every 25 frames during inference, and store 50 most recent tracked frames as the

	RC+IoU	DC+IoU	DC+Reg+IoU	DCC+NMS	DCC+IoU
AUC(%)	57.7	58.0	58.8	60.0	60.3

Table 1: Analysis of the deformable central correlation on the combined datasets. The baseline and further adding components are trained with pretrained backbone ResNet-50 and the balanced neck.

	Single Stage	BReg	BReg+EN	CReg	CReg+EN
NMS	60.0	60.0	60.1	60.3	60.7
IoU	60.3	60.4	60.5	60.6	60.8

Table 2: Comparison of different proposal refinement methods and analysis of the feature enhancement on the combined datasets.

template. As for offline training, we train our architecture effectively with a mixed training set created from TrackingNet, LaSOT and GOT10k. We sample various classes of images to form a mini-batch M_{train} and M_{test} to train the entire networks for 60 epochs with ADAM [Kingma and Ba, 2015] optimizer with learning rate decay of 0.2 every 15 epochs. We set $\sigma_c = 0.3$ for classification labels and set $\theta_{hi} = 0.6$ and $\theta_{lo} = 0.5$. In stage 2, we set the maximum of N_c to 25.

4 Experiments

Our new TLPG-tracker is implemented in Python with PyTorch and evaluated on five challenging tracking benchmarks: LaSOT [Fan *et al.*, 2019], VOT2018 [Kristan *et al.*, 2018], GOT10k [Huang *et al.*, 2018], TrackingNet [Muller *et al.*, 2018], OTB2015 [Wu *et al.*, 2015]. Employing with ResNet-50 as backbone, we achieve a tracking speed of 36 FPS with IoU-Net and 41 FPS with NMS on a single GeForce RTX 2080 Ti GPU.

4.1 Ablation Study

In this section, we analyze the effectiveness of three proposed components in TLPG-tracker with a combined dataset which contains 200 various video sequences randomly sampled from LaSOT, VOT2018 and OTB2015 datasets. The AUC metric is used in the evaluation.

Deformable Central Correlation. We investigate the impact of key aspects of the proposed deformable central correlation by incrementally adding each part of the module at a time which shows in Table 1. The baseline [Bhat *et al.*, 2019] contains an online learning regular convolution version filter (**RC**) with IoU-Net (**IoU**) as post-process. By replacing the filter with an online learning raw deformable convolution (**DC**), it gains a 0.3% AUC score on the combined dataset. Adding a proposal boxes regression loss (**Reg**) to the deformable convolution filter improves the AUC score by 0.8%. The deformable central correlation (**DCC**), which is performed only once during inference for fair comparisons, leads to a major improvement of 1.2% in AUC score with **NMS** and 1.5% in AUC score with IoU-Net. It shows that the combination of the localization, the representative target features and target shapes with the online learning loss fully exploit the correlation between target localization and shape generation. The deformable central correlation suits the post process well with a total improvement of 2.6% AUC score.

Center-bias Refinement with Feature Enhancement. We compare the proposed refinement method with classical bounding box regression. Here, NMS can indicate whether predicted scores of the proposal reflect the accuracy of both target localization and estimation while IoU-Net (**IoU**) showing the best accuracy of the proposal. We form a basic regression version (**BReg**) by replacing our center-bias refinement (**CReg**) with bounding box regression which refines the residual between proposal boxes and corresponding ground truth with extra convolution layers. Refinements are formed by using the feature enhancement (**EN**) or not, as shown in Table 2. Without enhancement, the center-bias refinement outperforms bounding box regression by 0.3% AUC score with NMS. The feature enhancement module gains up to 0.4% with center-bias refinement and NMS, but improves little with bounding box regression. It means that the center-bias refinement is compatible with NMS procedure.

	NO NECK	FPN	Balanced Neck
ResNet-50	59.5	60.1	60.8
DetNet-59	59.8	60.3	60.7

Table 3: Comparison of different neck modules on the combined datasets.

Balanced Neck. It is crucial to choose proper feature maps from backbone in tracking tasks. We compare our balanced neck module with FPN [Lin *et al.*, 2017] on two similar backbones, ResNet-50 and DetNet-59 [Li *et al.*, 2018b], used in image classification and object detection, to analyze the impacts of resolution and receptive fields. Backbone feature layers conv3, conv4, conv5 are involved. Using features of same spatial stride from neck modules or backbone as output feature maps, the performance of the Balanced Neck and FPN is shown in Table 3. The proposed balanced neck with ResNet-50 gains 1.3% AUC score improvement on the combined dataset and is 0.7% higher than FPN, which indicates features with balanced spatial details and sematic information are more suitable for object tracking. Note that DetNet-59 preforms better than ResNet-50 by 0.3% without a neck because it maintains high resolution in conv4 and conv5 while FPN aggregates the adjacent feature levels in a bottom-up manner. But they are less likely to provide balanced features.

4.2 State-of-the-art Comparison

The proposed **TLPG-Tracker**, which employs the backbone ResNet-50 and the post-process IoU-Net, is compared with the state-of-the-art methods including DiMP [Bhat *et al.*, 2019], SiamRPN++ [Li *et al.*, 2019], SiamMask [Wang *et al.*, 2019b], ATOM [Danelljan *et al.*, 2019], SPM [Wang *et al.*, 2019a], LADCF [Xu *et al.*, 2019], SiamGragh [Tu *et al.*, 2019], SiamRPN_R18 [Li *et al.*, 2018a], MFT[Kristan *et al.*, 2018], UPDT [Bhat *et al.*, 2018], StructSiam [Zhang *et al.*, 2018], MHIT [Bai *et al.*, 2018], VITAL [Song *et al.*, 2018], DRT [Sun *et al.*, 2018], DaSiamRPN [Dai *et al.*, 2017], BACF [Kiani Galoogahi *et al.*, 2017], ECO [Danelljan *et al.*, 2017], DSiam[Guo *et al.*, 2017], SiamFCv2 [Valmadre *et al.*, 2017], GOTURN [Held *et al.*, 2016], SiamFC [Bertinetto *et al.*, 2016], CCOT[Danelljan *et al.*, 2016], MDNet [Nam and Han, 2016] on following five challenging tracking benchmarks.

	ECO	DRT	UPDT	SiamMask	DaSiamRPN	MFT	LADCF	ATOM	SiamRPN++	DiMP	**TLPG**
EAO (↑)	0.281	0.356	0.378	0.380	0.384	0.385	0.389	0.401	0.414	0.440	**0.459**
Accuracy (↑)	0.483	0.519	0.536	0.609	0.586	0.505	0.503	0.590	0.600	0.597	**0.606**
Robustness (↓)	0.276	0.201	0.184	0.276	0.276	0.140	0.159	0.234	0.204	0.153	**0.149**

Table 4: State-of-the-art comparison on the VOT2018 dataset in terms of expected average overlap (EAO), robustness and accuracy.

	MDNet	ECO	CCOT	GOTURN	SiamFCv2	SiamRPN_R18	SPM	ATOM	DiMP	**TLPG**
AO(%)	29.9	31.6	32.5	34.7	37.4	48.3	51.3	55.6	61.1	**62.9**
$SR_{0.50}$(%)	30.3	30.9	32.8	37.5	40.4	58.1	59.3	63.4	71.7	**73.5**

Table 5: State-of-the-art comparison on the GOT10k test set in terms of average overlap (AO) and success rate (SR) at overlap threshold 0.5.

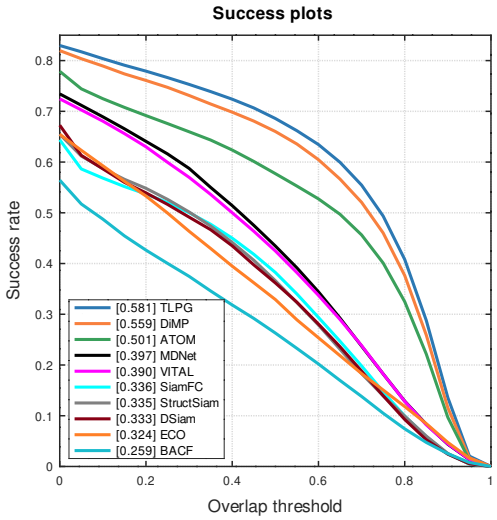

Figure 5: Success plots on the LaSOT test dataset.

LaSOT. The LaSOT dataset contains a large-scale, high-quality dense annotations with 1,400 videos in total. The evaluation is performed on the test set consisting 280 videos. The success plots are shown in Figure 5. The TLPG-Tracker, with an AUC score of 58.1%, outperforms the previous best tracker DiMP which employs online learning strategies with pretrained ResNet-50 features by 2.2% in success rate. The results demonstrate the powerful joint learning of target localization and proposal generation.

VOT2018. We evaluate our method on the VOT2018 challenge in terms of the measurement of robustness (failure rate) and accuracy (average overlap in the course of successful tracking) on the test split consisting 60 videos. Results are ranked by EAO (Expected Average Overlap) which are shown in Table 4. Previous results show that SiamMask is the most accurate of the short-term real-time challenge tracking results and DiMP is the beat in the comprehensive performance with EAO score of 0.440. With impressive robustness, our TLPG-Tracker obtains the best EAO score of 0.459 and the second accuracy score of 0.606.

GOT10k. GOT10k is a large tracking dataset containing over 10,000 videos which populate a majority of 563 target categories and over 80 motion patterns belong to real objects. Specially, all object classes between training videos and testing videos are non-overlapping except for the person class. We train our tracker only on the train split of GOT10k for fair comparisons. Results on the GOT10k test set are shown in Table 5. Our method outperforms the most progressive performance approaches DiMP with average overlap (AO) scores of 62.9% and success rates (SR) at thresholds 0.5 of 73.5%.

	UPDT	SPM	SiamGraph	SiamRPN++	ATOM	DiMP	**TLPG**
Precision(%)	55.7	66.1	63.8	69.4	64.8	68.7	**70.6**
Success(%)	61.1	71.2	70.9	73.3	70.3	74.0	**75.8**

Table 6: State-of-the-art comparison on the TrackingNet test dataset in terms of precision and success.

	ATOM	DiMP	SPM	ECO	SiamRPN++	MHIT	UPDT	**TLPG**
AUC(%)	66.9	68.4	68.7	69.1	69.6	69.8	70.2	**69.8**

Table 7: State-of-the-art comparison on the OTB2015 dataset in terms of area under the curve (AUC) score.

TrackingNet. TrackingNet is a huge dataset for target tracking, containing more than 30, 000 videos. As the results shown in Table 6, our TLPG-Tracker outperforms most advanced algorithms on TrackingNet test set, with the best precision rate of 70.6% and success score of 75.8%.

OTB2015. The standardized OTB2015 benchmark containing 100 videos with various challenging factors provides a fair testbed on robustness. Among all compared methods in Table 7, our TLPG-Tracker achieves a competitive result with an AUC score of 69.9% while UPDT obtaining the best performance with an AUC score of 70.2%.

5 Conclusion

We propose a novel two-stage tracking architecture that learn both target localization and proposal generation simultaneously by deformable central correlation in online training manner. The proposals are generated on discriminative features near target centers in stage 1 and refined by center-bias refinement to provide more accurate results. The model benefits from multi-level aggregation of fine-grained and sematic features in a neck and a feature enhancement module. With robust localization and accurate estimation, our TLPG-Tracker sets a new state-of-the-art on 5 datasets while operating at over 30 FPS.

References

[Bai *et al.*, 2018] Shuai Bai, Zhiqun He, Ting-Bing Xu, Zheng Zhu, Yuan Dong, and Hongliang Bai. Multi-hierarchical independent correlation filters for visual tracking. *arXiv preprint arXiv:1811.10302*, 2018.

[Bertinetto *et al.*, 2016] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H.S. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865, 2016.

[Bhat *et al.*, 2018] Goutam Bhat, Joakim Johnander, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Unveiling the Power of Deep Tracking. In *ECCV*, pages 493–509, 2018.

[Bhat *et al.*, 2019] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning Discriminative Model Prediction for Tracking. *arXiv preprint arXiv:1904.07220*, 2019.

[Bolme *et al.*, 2010] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *CVPR*, pages 2544–2550, 2010.

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

[Danelljan *et al.*, 2016] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *ECCV*, pages 472–488, 2016.

[Danelljan *et al.*, 2017] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, pages 6931–6939, 2017.

[Danelljan *et al.*, 2019] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. ATOM: Accurate Tracking by Overlap Maximization. In *CVPR*, 2019.

[Fan *et al.*, 2019] 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.

[Guo *et al.*, 2017] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *ICCV*, pages 1763–1771, 2017.

[Held *et al.*, 2016] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 fps with deep regression networks. In *ECCV*, pages 749–765, 2016.

[Huang *et al.*, 2018] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv preprint arXiv:1810.11981*, 2018.

[Jiang *et al.*, 2018] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, pages 784–799, 2018.

[Kiani Galoogahi *et al.*, 2017] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *ICCV*, pages 1135–1143, 2017.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kristan *et al.*, 2018] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Čehovin, Tomas Vojir, Goutam Bhat, Alan Lukežič, Abdelrahman Eldesokey, Gustavo Fernandez Dominguez, Alvaro Garcia-Martin, Álvaro Iglesias-Arias, A. Alatan, Abel Gonzalez-Garcia, Alfredo Petrosino, Alireza Memarmoghadam, Andrea Vedaldi, and Andrej Muhič. The sixth visual object tracking vot2018 challenge results. In *ECCV workshop*, pages 3–53, 2018.

[Li *et al.*, 2018a] 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.

[Li *et al.*, 2018b] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. DetNet: Design backbone for object detection. In *ECCV*, pages 334–350, 2018.

[Li *et al.*, 2019] 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.

[Lin *et al.*, 2017] Tsung Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[Muller *et al.*, 2018] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, pages 300–317, 2018.

[Nam and Han, 2016] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016.

[Song *et al.*, 2018] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, and Ming-Hsuan Yang. Vital: Visual tracking via adversarial learning. In *CVPR*, pages 8990–8999, 2018.

[Sun *et al.*, 2018] Chong Sun, Dong Wang, Huchuan Lu, and Ming-Hsuan Yang. Correlation tracking via joint discrimination and reliability learning. In *CVPR*, pages 489–497, 2018.

[Tu *et al.*, 2019] Zhengzheng Tu, Ajian Zhou, Bo Jiang, and Bin Luo. Visual object tracking via graph convolutional representation. In *ICMEW*, pages 234–239, 2019.

[Valmadre *et al.*, 2017] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 2805–2813, 2017.

[Wang *et al.*, 2019a] Guangting Wang, Chong Luo, Zhiwei Xiong, and Wenjun Zeng. SPM-Tracker: Series-Parallel Matching for Real-Time Visual Object Tracking. *arXiv preprint arXiv:1904.04452*, 2019.

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

[Wu *et al.*, 2015] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[Xu *et al.*, 2019] Tianyang Xu, Zhenhua Feng, Xiaojun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE Transactions on Image Processing*, 28(11):5596–5609, 2019.

[Yang *et al.*, 2019] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. RepPoints: Point Set Representation for Object Detection. *arXiv preprint arXiv:1904.11490*, 2019.

[Zhang *et al.*, 2018] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, pages 351–366, 2018.

[Zhu *et al.*, 2018] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *ECCV*, pages 103–119, 2018.

[Zhu *et al.*, 2019] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. *arXiv preprint arXiv:1904.05873*, 2019.