Neurocomputing 449 (2021) 48–60

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

AEVRNet: Adaptive exploration network with variance reduced optimization for visual tracking

Yuxiang Yang [a], Weiwei Xing [a,*], Dongdong Wang [b], Shunli Zhang [a], Qi Yu [a], Liqiang Wang [b]

[a] School of Software Engineering, Beijing Jiaotong University, Beijing 100044, China
[b] School of Computer Science, University of Central Florida, FL 32816, USA

ARTICLE INFO

Article history:
Received 27 October 2020
Revised 10 March 2021
Accepted 29 March 2021
Available online 9 April 2021
Communicated by Zidong Wang

Keywords:
Object tracking
Convolutional neural network
Reinforcement learning
Policy gradient
Adaptive exploration

ABSTRACT

For visual tracking methods based on reinforcement learning, action space determines the ability of exploration, which is crucial to model robustness. However, most trackers adopted simple strategies with action space, which will suffer local optima problem. To address this issue, a novel reinforcement learning based tracker called AEVRNet is proposed with non-convex optimization and effective action space exploration. Firstly, inspired by combinatorial upper confidence bound, we design an adaptive exploration strategy leveraging temporal and spatial knowledge to enhance effective action exploration and jump out of local optima. Secondly, we define the tracking problem as a non-convex problem and incorporate non-convex optimization in stochastic variance reduced gradient as backward propagation of our model, which can converge faster with lower loss. Thirdly, different from existing reinforcement learning based trackers using classification method to train model, we define a regression based action-reward loss function, which is more sensitive to aspects of the target states, e.g., the width and height of the target to further improve robustness. Extensive experiments on six benchmark datasets demonstrate that our proposed AEVRNet achieves favorable performance against the state-of-the-art reinforcement learning based methods.

1. Introduction

Visual tracking is one of the fundamental components in the field of computer vision, and has been intensively studied in video surveillance, intelligent transportation, unmanned aerial vehicles, and autonomous driving. Usually, its task involves automatically tracking the targets of interest in video sequences. Since the type and size of tracking object are uncertain, the robustness of tracker is an important index to measure its performance. However, it remains challenging to precisely tracking in complex scene, since many factors, like light changes, occlusion, motion blur, scale changes, and rotation to inner plane, will affect its performance.

With the great improvement of computing power, deep learning has been successfully applied to object tracking and achieves great success. Deep learning based methods take advantage of deep neural networks, such as, CNN [1–3], RNN [4,5], GANs [6–8], and Siamese network [9], to predict object position. These methods usually yield good robustness since tracking models are developed based on the training datasets with a large number of labeled images. However, the sampling and searching strategies they adopted without temporal and spatial knowledge of the object, and thus lead to inefficient and limited space searching during tracking process.

Recently, reinforcement learning (RL) based methods have been introduced into object tracking and achieve remarkable performance. ADNet [10], as one of the successful implementations, builds an appearance model with a supervised learning method and designs object motion model by policy gradient method. Actor-Critic [11], as another exemplary implementation, modifies the Actor-Critic framework to solve visual tracking problem based on continuous action space. Although they achieve better tracking performance compared to other RL-based methods, higher convergence cost affects their tracking efficiency and limited action space causes lower accuracy. For example, ADNet suffers frequent re-detection during online tracking and these corrections arise from local optima and limited action space. This case may become worse when object features significantly differ from the training datasets. Meanwhile, either Actor-Critic or ADNet shows high variance in

* Corresponding author.
E-mail address: wwxing@bjtu.edu.cn (W. Xing).

https://doi.org/10.1016/j.neucom.2021.03.118
0925-2312/

backward propagation, which reduces convergence speed and robustness of tracker.

In order to solve these problems, a novel regression RL-based tracker AEVRNet is proposed. To accelerate and reduce volatility during training and improve tracker robustness, non-convex optimized stochastic variance reduced gradient (SVRG) backward propagation is proposed to balance stochasticity induced noise and ensure fast convergence speed. For solving inefficient and limited tracking space searching problem, an adaptive exploration strategy is designed by combining spatio-temporal knowledge to enhance tracker exploration ability and helps escape local optima. A novel regression based action-loss function is defined to further improve the sensitivity of the target states in AEVRNet and reduce target loss caused by target states change. The performance of AEVRNet is evaluated on six benchmarks: OTB-2013 [12] with 50 sequences, OTB-100 [12] with 100 sequences, UAV123 [13] with 123 vehicle sequences, NFS [14] with 100 high frame rate sequences, TC128 [15] with 128 colorful video sequences and VOT16 with 61 sequences [16]. The results show that our tracker outperforms the state-of-the-art RL-based trackers.

The contributions of this paper include:

- A novel robust RL-based object tracker AEVRNet is proposed. To our knowledge, this is the first attempt to use non-convex optimized SVRG is designed for both deep neural network and policy gradient to accelerate model training and improve model robustness.
- An adaptive exploration method is designed to balance exploration and exploitation. By taking spatio-temporal knowledge into consideration, the proposed method can jump out of local optima and improve the tracking accuracy.
- A regression based action-reward loss function is defined to improve the robustness of RL-based trackers, which is more sensitive to aspects of the target states. Results of six benchmark datasets show that AEVRNet achieves favorable performance against the state-of-the-art RL-based methods in terms of accuracy and robustness.

The rest of this paper is organized as follows. We review related work in Section 2, and outline AEVRNet in Section 3. Section 4 illustrates experimental results including comparison with state-of-the-art methods. Section 5 draws conclusions.

2. Related work

2.1. Deep learning based methods

Based on well-developed deep neural networks, like CNN, efficient and accurate inter-frame algorithms are designed to address computer vision and pattern recognition problems. Deep learning methods in tracking can be classified into two categories. The first category uses the powerful representation capability of deep learning, which will improve the robustness of tracking. For example, Dong et al. [17] design a two-stage classifier to track the target in occlusion situation, and Ma et al. [18] use sparse represents to handle the target in motion blur situation. Shen et al. [19] adaptively refine the tracking targets and tracking boxes by introducing the minimum output sum of squared error filter. YCNN [20] proposes a two-flow CNN to measure the similarity between two image patches. Shen et al. [21] design submodular optimization to form the object trajectory in complex scene. Du et al. [22] propose tracking method based on LSTM to learn spatial–temporal of the tracking target. Ma et al. [23] train semi-supervised linear kernel classifiers for visual tracking with Fisher vectors. Scale adaptive tracking method is introduced into tracking by [24]. Based on their methods, Qi et al. [25] combine scale and state to present the tracking target. Huang et al. [26] capture the structure of the target by a part space with two online learned probabilities. Hu et al. [27] use both labeled and unlabeled samples to improve the robustness of model. CREST [28] and UCT [29] integrate discriminative correlation filter (DCF) processes with neural networks for end-to-end training. VITAL [6] introduces adversarial learning to improve tracking performance. However, they do not provide a powerful target motion strategy and the high robustness of tracker is based on extensive off-line training and complex networks, which is time-consuming and will limit the speed of online tracking.

The second category is Siamese network, which has shown great potential in tracking accuracy and speed. Siamese network compares the similarity between the search area and the object template in the first frame without update [9]. Based on that, SiameseFC [30] improves tracking performance using fully convolutional networks to search object. Shen et al. [31] design attention mechanism with Siamese network and improve the matching discrimination. Dong et al. [32] propose a quadruplet network to learn the relationship between samples and achieve better representation ability. SA-Siam [33] uses dual connection networks and channel awareness mechanisms to improve performance. Liang et al. [34] extract local semantic features with more fine-grained and partial information to solving drift problem. Dong et al. [35] propose a triplet loss to extract more discriminative deep features and Lu et al. [36] design a shrinkage loss to penalize the importance of easy samples. RASNet [37] introduces off-line trained general, target adapted residual, and channel favored feature attention methods into Siamese network. However, Siamese networks focus on utilizing the appearance information and pay less attention to the background information, which is crucial for discriminating the target from similar objects. To address these issues, our proposed method in this paper provides a regression based network considering both target and background information for visual tracking.

2.2. Reinforcement learning based methods

RL has been developed rapidly in recent years. As a sequential decision-making method, RL has successfully solved lots of problems in scientific research, engineering, liberal arts and other disciplines [38].

Recently, deep RL has been introduced to visual tracking. Compared with deep learning based methods, deep RL analyzes tracking problems more accurately by self-learning. Meanwhile, augmenting training samples improves the discriminative ability of tracker.

For example, HP [39] adopts a hyperparameter optimization method to learn appropriate hyperparameters, and designs a continuous deep Q-learning framework to track the object, an efficient heuristic strategy is also proposed by [40] to handle high dimensional state space and accelerate tracking. ADNet [10] builds an object appearance model with a supervised learning method, and implements policy gradient method to train motion model. EAST [41] improves the tracking efficiency using an off-line trained agent to determine optimal number of layers for motion prediction. The P-tracker [42] views object tracking process as a partially observable decision-making process (POMDP), and updates the model only when tracking drift occurs. This tracker uses an unlimited stream of Internet videos as training samples.

However, traditional RL-based trackers usually employ greedy method to explore action vectors and select the best action with the highest evaluation score as current solution. This method performs well on exploitation of the knowledge of current optimal actions, but suffers limited action space of exploration, and stuck in local optima [39,10]. Hence, instead of simple action exploitation strategy, we propose an adaptive exploration to expand the action space and balance of exploration and exploitation to improve tracking performance.

3. Adaptive exploration network with non-convex optimization for tracking

3.1. Overview

In this paper, we propose a novel robust RL-based tracker AEVR-Net. The tracking framework is shown in Fig. 1, which is divided into three stages: 1) off-line supervised training, 2) off-line RL training, and 3) online tracking. The model structure and the tracking problem definition based on reinforcement learning will be introduced in the following.

3.1.1. Off-line supervised training stage

For off-line supervised training stage, an initial model is trained with supervised learning based on non-convex optimization, and we obtain the supervised trained model. In this stage, we use supervised learning method to train off-line model on the ImageNet dataset [1]. By training with a large number of off-line training samples, the trained model can distinguish the tracking target and the background. Then, the trained model will be provided as the initial model for the next stage of RL learning training.

3.1.2. Off-line RL training stage

For the off-line RL training stage, non-convex optimization, regression based loss function and adaptive exploration methods are used to train the supervised trained model and obtain online tracking model. In this stage, the tracking model learns how to choose action during the tracking process and achieve better tracking results.The off-line RL training process is based on ALOV300 [43] dataset, which is same as ADNet.

3.1.3. Online tracking stage

For the online tracking stage, with the background and bounding box of previous frame, our tracker will predict the location and scale of the object in current frame by adaptive exploration, and tracker will be updated continuously. During the tracking process, the adaptive exploration will combine the spatio-temporal information of the current tracking target to give a better choice of action, making the tracking model can jump out of a local optimum solution.

3.2. Problem definition

Considering object tracking as a sequential decision problem, we achieve reinforcement learning by introduce Markov decision process (MDP) to define the tracking problem. MDP consists of four major entities: action $a \in A$, state $s \in S$, state transition function $s\prime = f(s,a)$, reward $r(s,a)$ and the four entities can be specified as follows.

Action: Action $a = (\Delta x^t, \Delta y^t)$ is defined by the change of bounding boxes involving 11 object move actions (i.e., left, right, up, down, double left, double right, double up, and double down), scale changes (scale up and scale down), and stop. Each action is encoded by one 11-dimensional vector with one-hot form. In detailed, $\Delta x^t = \alpha w^t, \Delta y^t = \alpha h^t, \alpha$ is set as 0.3 as ADNet.

State: State s_t is described by the information within bounding box p_t and the dynamics of actions denoted by action dynamics vector d_t. p_t is consist of $b_t = (x^t, y^t, w^t, h^t)$, which means the center position and the width and the height of the tracking box, respectively. In detailed, a per-processing function $s = \phi(b_t, F)$ is defined to crop the image patch within the bounding box b_t in a given frame F.

State transition function: State transition function is formulated with horizontal and vertical change. The tracker will move the target bounding box according to new position according to the the prediction action. The b_t can be changed to b_{t+1} by a_{t+1}, and the transition can be formulated as $(x^t \pm \Delta x^t, y^t \pm \Delta y^t, h^t \pm \Delta x^t, w^t \pm \Delta y^t)$. Different from the existing RL-based trackers using simple strategies to build action space, we innovatively design an adaptive exploration strategy combined with temporal and spatial information to enhance exploration of our tracker, which can successfully escape local optima.

Reward: The reward function $r(s,a)$ means the improvement of tracking accuracy by taking action a and transfer state s into state s'. It is by the overlap ratio of the predicted bounding box b_t and the ground truth G, as Eq. (10).

3.3. Tracking model structure

In the proposed tracking framework, we follow the popular object tracking model structure. As shown in Fig. 2, the backbone structure is designed with three convolutional layers of VGG-M and two fully-connected layers with ReLU activation. The last layer

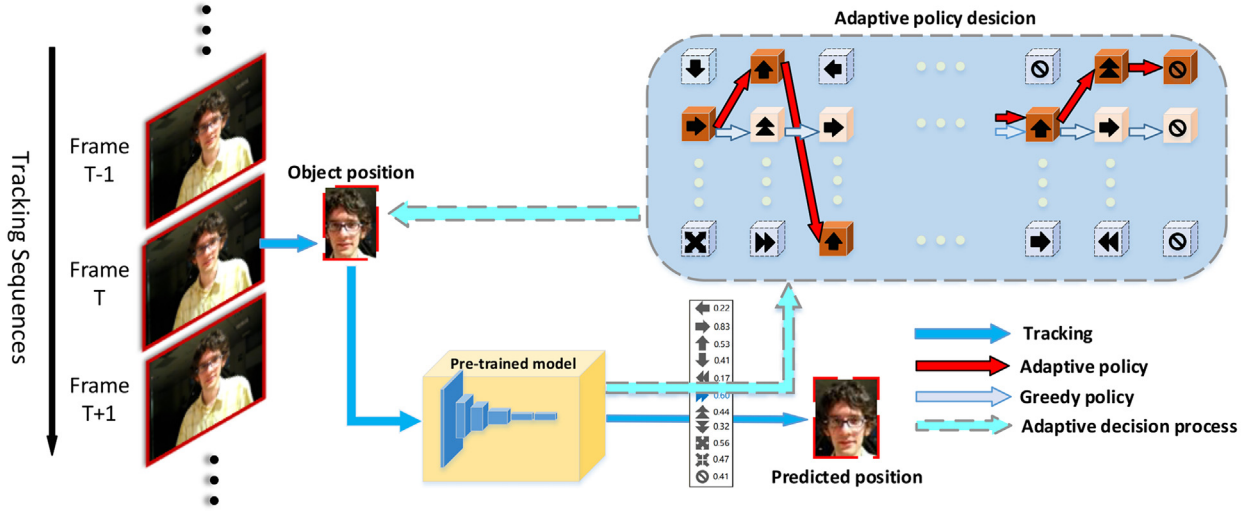


Fig. 1. The online tracking framework of our proposed tracker AEVRNet. Assume default greedy search yields the policy plan along the direction of light blue arrows. The red arrows show the adaptive exploration policy decision.

is connected to a vector consisting of action probabilities and confidence scores. In order to further improve the training convergence speed and model accuracy of RL-trackers training methods, we innovatively propose non-convex optimized SVRG as backward propagation to relieve local optima problem for tracking problem. Moreover, an action-reward loss function is designed with regression to train AEVRNet, which honors Intersection Over Union (IoU) between the estimation and ground truth bounding boxes. Different from existing RL-based trackers using classification, regression is more sensitive to the target states, such as the width and height of the target. The model can learn more scale change information and reduce target loss caused by extra interference information, which can further improve the accuracy of the proposed tracker. The main contributions and details are described as follows.

3.4. Off-line pre-training

3.4.1. Non-convex optimized variance reduced backward propagation

Supervised learning for object tracking problem can be formulated into composite optimization problem:

$$\min_{x\in\mathbb{R}} F(x) \triangleq f(x) + g(x) \tag{1}$$

where $f(x)$ is a smooth function and $g(x)$ is referred to as regularizer. Currently, majority of tracking problems are solved by SGD method [44] with the backward propagation as Eq. (2), where θ is model parameter, B denotes mini-batch, g is regularier, η is learning rate, and t refers to update iteration.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t[\frac{1}{B}\sum_{i=1}^{B}\nabla f_i(\boldsymbol{\theta}_t) + \nabla g(\boldsymbol{\theta}_t)]. \tag{2}$$

However, SGD usually takes much time to converge because of the large variance of $\nabla f_i(\boldsymbol{\theta}_t)$ during random sampling [45]. Dong et at. [39] use sequence based hyperparameter optimization method to improve the discrimination ability of model. Anschel et al. [46] average previously estimates and reduce the approximation error variance. Wu et al. [47] propose a triplet-average policy gradient to reduce the estimation bias. Pourchot et al. [48] combine the cross-entropy and twin delayed deep deterministic policy gradient to improve the robustness of model. In order to improve the accuracy and efficiency of our proposed AEVRNet, a non-convex SVRG backward propagation is introduced to instead of SGD as shown in Eq. 3.

Supposing the current training epoch is s, and we will develop a snapshot for the model parameters obtained from last training epoch $\boldsymbol{\theta}^{s-1}$. Given the current sampled data i, the previous gradient $\nabla f_i(\boldsymbol{\theta}^{s-1})$ is calculated based on the snapshot model parameters and the average gradient $\hat{\mu}$ across all data in one epoch is calculated with the snapshot model parameters. The difference between them is used to adjust current calculated gradient, thus reducing model updating variance, which is shown in Eq. 3.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \cdot [\hat{\mu} + \frac{1}{B}\sum_{i=1}^{B}(\nabla f_i(\boldsymbol{\theta}_t^s) - \nabla f_i(\boldsymbol{\theta}^{s-1})) + \nabla g(\boldsymbol{\theta}_t^s)]. \tag{3}$$

Furthermore, visual tracking is a non-convex problem and the backward propagation without non-convex optimization may severely suffer local optima and cause tracking failure. Therefore, we conduct non-convex optimization with stochastic process to alleviate premature convergence to local optima and the details are shown in Algorithm 1.

Algorithm 1. Non-convex SVRG Optimization for Supervised Learning Training Process.

Input: a set of images D_N, number of epochs S, epoch size m, step size η, initial parameter $\boldsymbol{\theta}_m^0 := \hat{\boldsymbol{\theta}}^0$.

for $s = 0$ to $S-1$ **do**
  $\boldsymbol{\theta}_0^{s+1} := \hat{\boldsymbol{\theta}}^s = \hat{\boldsymbol{\theta}}_m^s$
  $\hat{\mu} = \nabla f(\hat{\boldsymbol{\theta}}^s)$
  for $t = 0$ to $m-1$ **do**
    $x_B \sim U(D_N)$
    $v_t^{s+1} = \hat{\mu} + \frac{1}{B}\sum_{i=0}^{B-1}(\nabla f_i(x|\boldsymbol{\theta}_t^{s+1}) - \nabla f_i(x|\hat{\boldsymbol{\theta}}^s))$
    $\boldsymbol{\theta}_{t+1}^{s+1} = \boldsymbol{\theta}_t^{s+1} + \eta v_t^{s+1}$
  end for
end for
return $\boldsymbol{\theta}_t^s$ for a random pair $(s,t) \in \{[0,S-1]\times[0,m-1]\}$

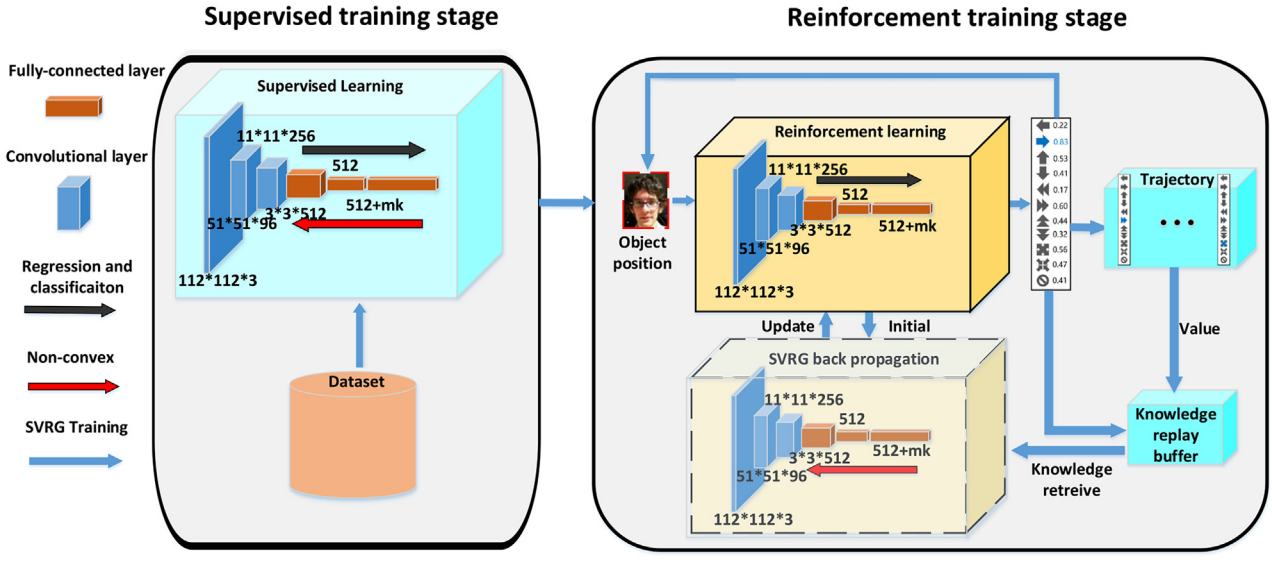


Fig. 2. The pre-training process of our proposed method AEVRNet. The pipeline starts from image database, and then, goes through supervised learning training, and next, experiences fine-tuning with RL. The forward propagation is conducted based on regression loss function. The backward propagation is accomplished with non-convex SVRG method.

Same as supervised learning, non-convex optimized SVRG backward propagation is also introduced into RL training process to alleviate convergence to local optima. Inspired by [49], the period gradient will adjust current calculated gradient and reducing model updating variance. When solving policy gradient problem, **Algorithm** 1 will be subjected to bias and a correction term ω is employed during gradient projection. This term ω is calculated by $\omega(\tau|\theta_t,\hat{\theta}) = \frac{p(\tau|\theta)}{p(\tau|\theta_t)}$ with importance weighting from policy snapshot. The details are shown in Algorithm 2.

As far as we know, it is the first time to introduce SVRG backward propagation into object tracking. In particular, we innovatively designed non-convex optimized SVRG backward propagation for both supervised learning and RL visual tracking process, as shown in Fig. 2.

Algorithm 2. Non-convex SVRG Optimization for Policy Gradient Tracking Training Process.

Input: a set of images D_N, number of epochs S, epoch size m, step size η, batch size N, mini-batch size B, gradient estimator g, initial parameter $\theta_m^0 := \hat{\theta}^0$.

for $s = 0$ to $S-1$ **do**
  $\theta_0^{s+1} := \hat{\theta}^s = \hat{\theta}_m^s$
  Sample N trajectories$\{\tau_i\}$
  $\hat{\mu} = \hat{\nabla}_N J(\hat{\theta}^s)$
  for $t = 0$ to $m-1$ **do**
    Sample B trajectories$\{\tau_i\}$ from $p(.|\hat{\theta}_t^{s+1})$
    $\boldsymbol{c}_t^{s+1} = \frac{1}{B}\sum_{i=0}^{B-1}(\nabla f(\tau_i|\theta_t^{s+1}) - \omega(\tau_i|\theta_t^{s+1},\hat{\theta}^s)\nabla f(\tau_i|\hat{\theta}^s))$
    $v_t^{s+1} = \hat{\mu} + \boldsymbol{c}_t^{s+1}$
    $\theta_{t+1}^{s+1} = \theta_t^{s+1} + \eta v_t^{s+1}$
  end for
end for
return θ_t^s for a random pair $(s,t) \in \{[0,S-1]\times[0,m-1]\}$

3.4.2. Adaptive exploration based on combinatorial upper confidence bound

RL-based object tracking methods usually suffer the issue of local optima due to limited action space. When model reaches a local optima, the limited action space prevents the model from jumping out of the local optima and causes tracking failure. However, existing RL-based trackers try to solve this problem with simple action space searching strategy, which may still stuck in local optima. Hence, we propose an adaptive exploration method that employs spatio-temporal information to optimize action space search and can jump out of local optima to find a better solution. This method expends the action space with spatio-temporal information, and also improves the tracking performance by leveraging the balance between exploitation and exploration.

The adaptive exploration strategy is formulated with policy gradient and combinatorial upper confidence bound (CUCB) [50] to expand the action space with spatio-temporal information and then the action obtained by adaptive exploration strategy will fine-tune deep neural network to jump out of local optima and find a better solution.

Algorithm 3. Adaptive Exploration for Policy Search.

Input: arbitrarily initialized θ, frame number T, current frame t_0, step size η, episode length τ, update interval k, action set $\boldsymbol{A}$, and episode set $\varnothing$.

for $t = t_0$ to t_τ by k **do**
  if $t_0 > T$ **then**
    $a_t = \arg\max_a Q_t(a)$
  else
    $a_t \sim CUCB[\boldsymbol{A_t}]$
  end if
end for
for each episode $\{s_{t_0}, a_{t_0}, r_{t_1}, \ldots, s_{t_{\tau-1}}, a_{t_{\tau-1}}, r_{t_\tau}\} \sim \pi_\theta$ **do**
  for $t = t_0$ to t_τ **do**
    $\theta \leftarrow \theta + \eta\nabla_\theta \log \pi_\theta(s_t,a_t)Q^{\pi_\theta}(s_t,a_t)$
  end for
end for
return θ

Our adaptive exploration combines exploration actions into sequential exploitation process as shown in Fig. 1. Generally, stochasticity can expand the action search space, and can jump out of local optima. However, random perturbation may only jump out of local optima without find a better solution, which increases the risk of target lose due to stochasticity perturbation. To address this issue, the prior spatio-temporal knowledge of action explorations is adopted to improve the quality of action space search. While introducing stochasticity makes the model can jump out of local optima, it can also constrain stochasticity through the spatio-temporal information of previous tracking results to ensure that the model can find a better solution. The optimization process is summarized in Algorithm 3.

The adaptive exploration is initialized with the default greedy method. When a new frame comes, the tracking model will calculate the score for each action. Given the score set, the action with maximum score will be chosen for optimal policy solution and bounding box projection (shown by Eq. 4). After initialization, the tracking process will be warmed up with several frames.

$$A_t = \arg\max_a Q_t(a) \tag{4}$$

where $Q_t(a)$ denotes the score of an action and t is current frame number.

After warming up period, the action space search strategy will be replaced by CUCB, and X means the warm up stage flag, which is shown as Eq. (5), and the designed CUCB is shown as **Algorithm** 4. In line 4, $\hat{\mu}_i$ means the current average value of action i. t means the total number of the current selection, $\sqrt{\frac{3\ln t}{2M_i}}$ is the standard deviation of $\hat{\mu}_i$ and is used to update $\bar{\mu}_i$. It shows that as the number of trials for each action increases, the confidence interval becomes narrower and the probability of action is more certain. If the mean value of the action is greater, the greater chance of being selected, and if the mean value of the action is smaller, the less chance it will be selected. In line 5, the $Q_t(a_i$ of each action is added with the corresponding $\bar{\mu}_i$, and the action with maximum $Q_t(a_i$ will be selected as the A_i.

Algorithm 4. Combinatorial Upper Confidence Bound.

Input: action i, the total number of times action i is played in action memory M_i, the mean of all outcomes scores of action i observed in action memory $\hat{\mu}_i$.

For each action i, play A_i and update variables M_i and $\hat{\mu}_i$.
while true **do**
  $t \leftarrow t+1$
  For each action i, set $\bar{\mu}_i = \hat{\mu}_i + \sqrt{\frac{3\ln t}{2M_i}}$
  $A_t = \arg\max_a(Q_t(a_i) + \bar{\mu}_i)$
  Play action A_t and update all M_i and $\hat{\mu}_i$
end while

During the tracking process, the frame number T is an important hyper parameter and is used to control the beginning of adaptive exploration. When T is not smaller than a specific number, adaptive exploration system is activated. Otherwise, default greedy-search is activated. We test T with different value on OTB2013, which is detailed in Section 4, and the tracker performs better when $T = 30$. The tracking failure score is used for cost function and optimal hyper parameter is achieved after several iterations.

$$A_t = \begin{cases} CUCB(a), & \text{if } T \geqslant 30 \\ \arg\max_a Q_t(a), & \text{otherwise.} \end{cases} \tag{5}$$

3.4.3. Regression based training

The first stage is to pre-train tracker with supervised learning. The existing RL-based trackers, like ADNet or ACT, formulate action-reward function with classification setting. This formulation requires the location of the target to be converted into a marked training sample. However, spatial continuous information is lost during this conversion. To preserve this important information, we propose a regression training method for forward propagation to train the proposed model. We innovatively designed a regression based action-reward loss function for RL-based tracker. Instead of discrete binary reward function, a continuous function is developed to map IoU with reward, as shown in Eq. (7), and then, reward contains more detailed information of target.

The training dataset consists of image patches $\{p_j\}$, action labels $\{o_j^{(act)}\}$, and regression value $\{r_j^{(reg)}\}$. During training, the action dynamics vector $\{d_j\}$ is set to zero. The ground truth patch position, size and image, are provided. A sample patch p_j is generated around the ground truth with Gaussian importance sampling and its corresponding action $o_j^{(act)}$ is assigned by,

$$o_j^{(act)} = \arg\max_a IoU(\bar{f}(p_j, a), G) \tag{6}$$

where $\bar{f}(p_j, a)$ denotes the patch moved from p_j by action a and G means the ground truth patch. In order to make full use of the object information, the corresponding action regression value $r_j^{(reg)}$ to p_j is innovatively defined as follows,

$$r_j^{(reg)} = IoU(p_j, G). \tag{7}$$

Different with the existing RL-based trackers using the classification method to train the tracker, which is not sensitive to the target deformation and leading to target loss. We propose and define a regression based action-reward loss function, which is more sensitive to aspects of the target states, e.g., the width and height of the target and reduce tracking failure due to target deformation.

A training batch consists of the randomly selected training samples $\{(p_j, o_j^{act}, r_j^{reg})\}_{j=1}^m$. The proposed network (W_{SL}) minimizes the multi-task loss function by non-convex optimized SVRG. The multi-task loss function is defined by minimizing the following loss L_{SL},

$$L_{SL} = \frac{1}{m}\sum_{j=1}^{m} L_1(o_j^{(act)}, \hat{o}_j{}^{(act)}) + \frac{1}{m}\sum_{j=1}^{m} L_2(r_j^{(reg)}, \hat{r}_j{}^{(reg)}) \tag{8}$$

where m denotes the size of patch batch, L_1 denotes the cross-entropy loss, L_2 denotes the squared loss, and $\hat{o}_j{}^{(act)}$ and $\hat{r}_j{}^{(reg)}$ denote the predicted action and corresponding action regression IoU value, respectively.

The second stage is to pre-train tracker with RL. The network is fine-tuned by policy gradient approach and uses the same parameters of W_{SL} as the initial network parameters. The tracking process has sequential states $s_{t,l}$, the corresponding action $a_{t,l}$ and the reward function $r(s_{t,l})$. $a_{t,l}$ is defined by,

$$a_{t,l} = \arg\max_a p(a|s_{t,l}; W_{RL}) \tag{9}$$

where $p(a|s_{t,l})$ means conditional action probability and the reward function $r(s_{t,l})$ is as follows,

$$r(s_{t,l}) = \begin{cases} 1, & if\ IoU(b_T, G) > 0.7 \\ -1, & otherwise \end{cases} \tag{10}$$

where b_T means the terminal patch position.

Same as other trackers, we use the first six layers of the network to training. The parameters $(w_1, \ldots, w_6)$ in W_{RL} are updated by non-convex optimized SVRG to maximize the tracking scores as follows,

$$\Delta W_{RL} \propto \sum_{l=1}^{L}\sum_{t=1}^{T_l} \frac{\delta \log p(a_{t,l}|s_{t,l}; W_{RL})}{\delta W_{RL}} Z_{t,l} \tag{11}$$

where $Z_{t,l} = r(s_{t,l})$ means the reward, L is training frames, and T_l is steps during the $l-th$ frame.

3.5. Online tracking and update

After the first and second pre-training stages, the pre-trained tracker will track and be updated during online visual tracking for the third stage. For each frame, our method chooses the position with maximum score given by adaptive exploration strategy as the estimated object position. Then tracker will be updated with samples by Gaussian sampling around the predicted location. We only fine-tune the fc layers $w_4, \ldots, w_7$ instead of all layers, for the fc layers would have the video specific knowledge while convolutional layers would have generic tracking information. The tracking framework is shown in Algorithm 5.

Algorithm 5. Framework of Our Proposed Method for Online Tracking.

Input: initial object position P_0.
Output: estimated object position $P_t = (x_t, y_t, w_t, h_t)$;
1: Generate samples in the first frame to update all fully-connected layers in network;
2: **repeat**
3: Extract features from (x_{t-1}, y_{t-1});
4: **repeat**
5: Compute scores for 11 actions and choose action with adaptive policy exploration;
6: Move the bounding box with the selected action and add the selected into the action sequence;
7: Extract features from the bounding box;
8: **until** The selected action is a stop action;
9: Compute score of the bounding box;
10: **if** score<-0.5 **then**
11: Use re-detection module to find a position with a higher score around the bounding box;
12: **end if**
13: Update the network by the predicted position $P_t = (x_t, y_t, w_t, h_t)$ and action sequence;
14: **until** End of video sequence;

4. Experiments

Our proposed AEVRNet is implemented in MATLAB 2017b with MatConvNet toolbox, which runs on a PC with a 4-cores 4.2 GHz

Intel 7700k CPU and an NVIDIA 2080Ti GPU with 11G memory. During off-line training, we used ALOV300 [51] as the training dataset, which is the same as ADNet. At the stage of online tracking, only fully-connected layers are fine-tuned. Concerning sample generation, 150 negative and 200 positive samples are generated from the first frame. After the first frame, 15 negative and 20 positive samples are generated when tracking is successful, and 512 samples are generated in the re-detection model when tracking fails. In terms of adaptive exploration, T is set to 30, which means the CUCB strategy starts at the 31st frame.

4.1. Evaluation on OTB

Our proposed method is evaluated with OTB dataset, which is a popular benchmark dataset. The tracking performance is also compared of another nine state-of-the-art trackers, including ECO [52], MDNet [53], ADNet, DeepSRDCF [54], CF2 [55], HDT [56], SRDCFdecon [57], MEEM [58], and KCF [59]. These methods can be classified into CF based methods, deep learning based methods, and RL-based methods. The experiments are carefully designed based on the same protocols and the same parameters.

The selected OTB datasets include OTB-50, OTB-100, and OTB-2013. Fig. 4 shows the tracking results of all trackers under one-pass evaluation(OPE) on these datasets. The performance of our proposed method is shown in Fig. 4, exhibits high precision and success rate, and the state-of-the-art trackers are beaten. The precision of our tracker is 88.6%, 90.9%, and 95.3% on OTB-50, OTB-100, and OTB-2013, receptively. It is shown that the proposed method yields higher precision than ECO and MDNet in OTB-2013. In terms of OTB-50 and OTB-100, the precision of our method is comparable with ECO and MDNet. In addition, compared with ADNet, our method runs significantly faster and accurately.

To further analyze the performance of the proposed method, we choose more state-of-the-art methods for comparison in OTB-100, including PG-Net [60], Siam R-CNN [61], GradNet [62], DiMP [63], TADT [64], C-RPN [65], DAT [66], SPM [67]. The results are shown in 0Table 1. As shown in the table, the proposed method performs favorably against other state-of-the-art methods.

4.2. Quantitative evaluation

OTB divides the video sequences into 11 attributes (e.g., fast motion, occlusion, scale variation and illumination variation) and those attributes in OTB benchmark are also analyzed. Fig. 3 lists

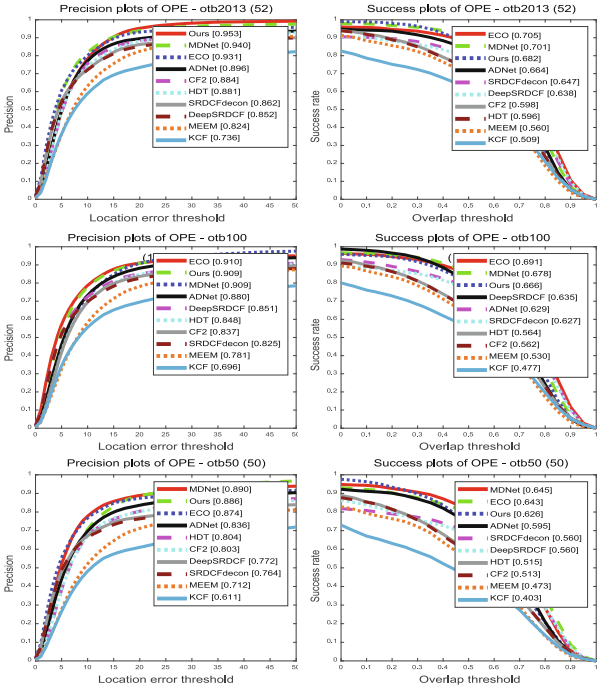


Fig. 4. Precision and success plots using the one-pass evaluation(OPE) over OTB-100, OTB-50, and OTB-2013 benchmarks. The legend of location error precision contains threshold score at 20 pixels for each tracker. The performance of AEVRNet is favorably against the state-of-the-art trackers.

the results from all trackers based on eight main video attributes of OPE in OTB-100. Our method AEVRNet still better performs on illumination variation, low resolution, and background clutter. Compared with ADNet, AEVRNet outperforms in scale variation, in-plane rotation by 2.6% and 4.1%, receptively. As our method use regression instead of classification method, which is more sensitive to aspects of the target states, e.g., the width and height of the target. Compared with ECO and MDNet, our proposed AEVRNet uses adaptive exploration to enhance larger action space and have chance to jump out of local optima. Meanwhile, the performance of the proposed method in fast motion slightly lower than some state-of-the-art trackers. This result is relevant to feature extraction of deep neural network. Since we use less layers in our con-

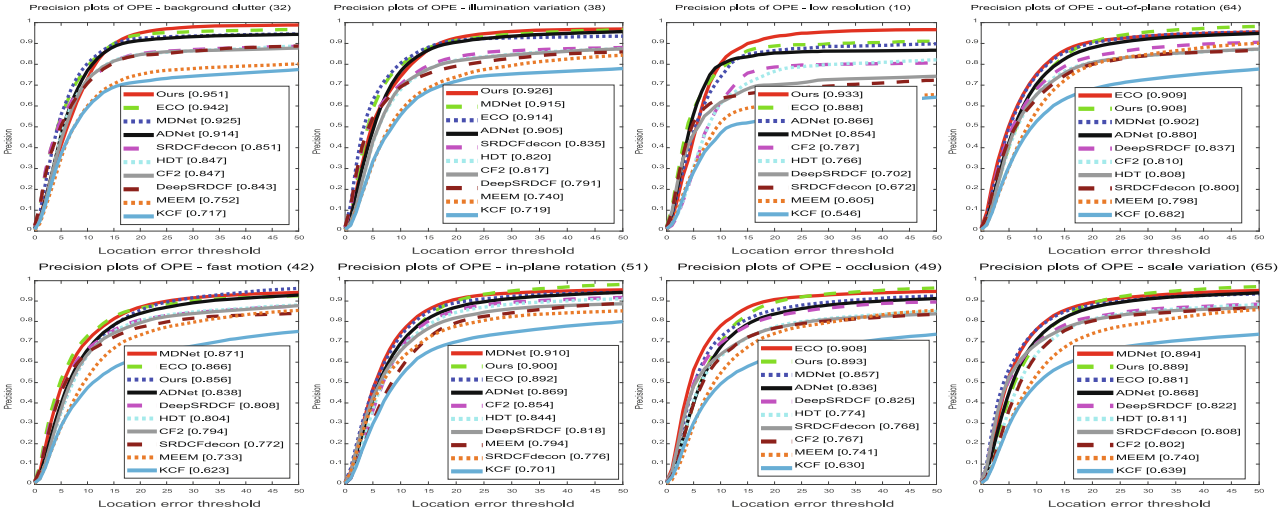


Fig. 3. Precision plots over eight tracking challenges. The scores are obtained with a threshold of 20 pixels for each tracker. Our proposed method AEVRNet performs favorably against the state-of-the-art trackers on these eight challenging attributes.

Table 1
Comparison with the state-of-the-art trackers on the OTB-100 dataset. The results are presented in terms of precision. The best score is highlighted in bold.

Tracker	Ours	PG-Net	Siam R-CNN	GradNet	DiMP
Precision	**0.909**	0.892	0.891	0.861	0.901
Tracker	TADT	C-RPN	DAT	SPM	SiamFC++
Precision	0.866	0.875	0.903	0.899	0.855

structed network to accelerate tracking, which lower the discriminative ability of feature.

4.3. Qualitative evaluation

Fig. 5 shows tracking results of several top tracking methods: MDNet, KCF, ADNet, CF2, together with our proposed method on seven challenging sequences. CF2 performs well in rotation and deformation conditions (*Diving* and *MotorRolling*), but misses the object when fast motion and large-scale variation occur (*Biker*, *Bird2*, and *Matrix*), because it has no re-detection module. KCF uses only HOG feature to represent the object, as a result, it can track fast but cannot fully describe the object, which leads to object missing. It also fails to track the object when heavy occlusion and background clutter occur (*Bird2*, *MotorRolling*, and *Matrix*). ADNet based on RL performs well in rotation and scale variation conditions (*Diving*, *Walking2*, and *MotorRolling*). However, for fast motion and heavy occlusion conditions, it may miss the object (*Biker*, *Matrix*, and *Bird2*) as its greedy strategy may not jump out of local optima. MDNet performs well in rotation, fast motion and occlusion conditions (*MotorRolling*, *Bird2*, and *Diving*) with multi-domain theory, but when background clutter and large-scale variation situations occur (*Biker*, *CarScale*, and *Matrix*), its performance is less accurate because the tracker cannot follow the object well under fast appearance changes.

The proposed method performs well for two main reasons. Firstly, the regression training method is more coupled to tracking problem. Different with the existing RL-based trackers using the classification method to train the tracker, which is not sensitive to the target deformation and leading to target loss. Our proposed method pays more attention to learn the target states, e.g., the width and height of the target. The results show that our proposed

Fig. 5. Qualitative evaluation of our method, MDNet, KCF, ADNet, and CF2 on seven challenging sequences, *Biker*, *Bird2*, *CarScale*, *Diving*, *Walking*, *MotorRolling*, and *Matrix*.

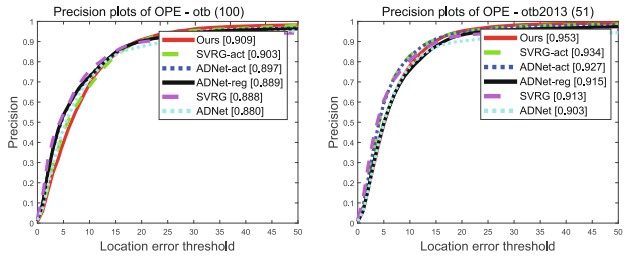


Fig. 6. Precision plots of different component of our proposed AEVRNet on OTB-100 and OTB-2013, which shows the improvement of each component of AEVRNet.

AEVRNet performs well in deformations, rotations, and scale variation conditions (*CarScale*, *Walking2*, *Diving*, and *MotorRolling*). Especially for *CarScale*, our method adjusts almost perfectly to the scale variation of the ground truth, much better than other methods. Secondly, the adaptive exploration method can enable the proposed tracker to expand the action space and jump out of local optima. Furthermore, the results show that our method performs well in occlusion, background clutter, and fast motion (*Biker*, *Bird2*, and *Matrix*). It performs better than ADNet in all aforementioned seven challenging sequences.

4.4. Ablation analysis

To further analyze the contribution of each component in the model, we evaluate different variations of our method on OTB-100 and OTB-2013. Here, ADNet is used as a baseline. "SVRG" denotes the case that the proposed method is only with non-SVRG to do off-line training and online tracking. "SVRG-Action" denotes the case that the proposed method used both non-SVRG and adaptive exploration for object tracking. "Ours" denotes the case that the proposed method used all of non-SVRG, adaptive exploration and regression based training.

The performance of all those variations is shown in Fig. 6. It is obtained that every single component can improve performance of the proposed method. For adaptive exploration combines temporal and spatial relations to solve the problem of RL-based visual tracking during online tracking, which can enhance exploration and successfully escapes local optima. The results show that it gains 2.9% and 5.0% improvements on two datasets. The regression reduces information loss and improves the robustness of the proposed method.

4.4.1. Non-convex optimized stochastic variance reduced gradient backward propagation

We analyze the impact of the proposed non-convex SVRG and trains 120 epochs, which is same as the ADNet settings. From Table 2 we can find that non-convex optimized SVRG (non-SVRG) can converge fast with lower loss on training and test datasets by 0.017 and 0.027, respectively. Because non-SVRG uses the optimal solution of the current epoch of training to initial the parameter of next epoch. Compared with SGD using the random parameter as the initial parameter, the model training can be accelerated. The online tracking results are shown in Fig. 6, the non-SVRG can improve the baseline's precision by 0.8% and 0.6% on two datasets. To further analyze the effective of the proposed non-convex SVRG, we apply it to famous deep learning based trackers ECO and MDNet. As shown in Table 3, the non-SVRG can gain 0.6% and 0.5% improvement on ECO and MDNet, respectively. It is mainly benefited from the robust model trained by non-SVRG can convergence faster with lower loss and improves the accuracy of the proposed method.

4.4.2. Adaptive exploration based on CUCB

We also analyze the impact of the hyper parameter T in adaptive exploration. The hyper parameter T is tested on OTB-2013 by different values from 0 to 50, and the proposed method performs better when T is 30 as shown in Table 4. That means when the tracker is robust to the object, the adaptive exploration can enhance the exploration ability better. Adaptive exploration combines temporal and spatial information to solve action space selection, which can enhance exploration and successfully escapes local optima. The results shows that it gains 2.3% and 3.1% improvements on two datasets shown in Fig. 6. As shown in Fig. 7, the adaptive exploration can effectively alleviate target loss in occlusion and blur. To further analyse the effect to action selection of μ, we design three action selection methods on OTB-100. I method chooses action with greedy, II method chooses action only with $\hat{\mu}_i$, and III method chooses with $\hat{\mu}_i + \sqrt{\frac{3\ln t}{2M_i}}$, the results are shown in Table 5. The results show that compared to the greedy method, the current average value $\hat{\mu}_i$ can obtain 1.1% and 0.8% improvement on precision and AUC. When $\hat{\mu}_i$ is combined with the standard deviation $\sqrt{\frac{3\ln t}{2M_i}}$, the results can be improved by 0.6% and 0.5% on precision and AUC, respectively.

4.4.3. Regression based training

We analyze the impact of regression based training. Fig. 8 shows that the regression is sensitive to different aspects of the target states, e.g., the width and height of the target, which can help tracker predict position more accurate around the tracking object. Fig. 6 shows that it gains 0.6% and 2.0% improvements on two datasets, since our model is sensitive to aspects of the target states, e.g., the width and height of the target and reduces interference information passed to tracker. Furthermore, the regression based training can also reduce re-detection frequency by 30% on average when blur occurs. To further analyse the effective of the proposed the regression based action-reward loss, we apply it to another reinforcement learning based tracker ACT, the experiment results are shown in Table 6. We can find the proposed method obtains 1.2% and 1.3% improvement in precision on OTB-100 and OTB-2013 datasets, and also improve the AUC rate, respectively. It is mainly because the proposed method is more sensitive to different aspects of the target.

Table 2
The training loss and test error of pre-training for our proposed method. The non-SVRG method performs better than SGD method, which convergences faster with lower loss and error.

Epoch		1	30	60	90	120
Training loss	SGD	1.753	1.578	1.569	1.562	1.558
	non-SVRG	1.748	1.559	1.550	1.545	1.541
Test error	SGD	0.543	0.436	0.429	0.423	0.419
	non-SVRG	0.540	0.407	0.401	0.396	0.392

Table 3
The precision results of the baseline, ECO, and MDNet with non-SVRG on OTB-100. The results show that the non-SVRG can be applied to other deep learning based trackers and improve their training process.

		Baseline	ECO	MDNet
Precision	SGD	0.880	0.910	0.909
	non-SVRG	0.888	0.916	0.914

Table 4
The average precision results on OTB-2013 dataset. The best scores are highlighted in bold. Our proposed method performs best when T = 30.

	T = 0	T = 10	T = 20	T = 30	T = 40	T = 50
Precision	0.920	0.926	0.935	**0.953**	0.941	0.933

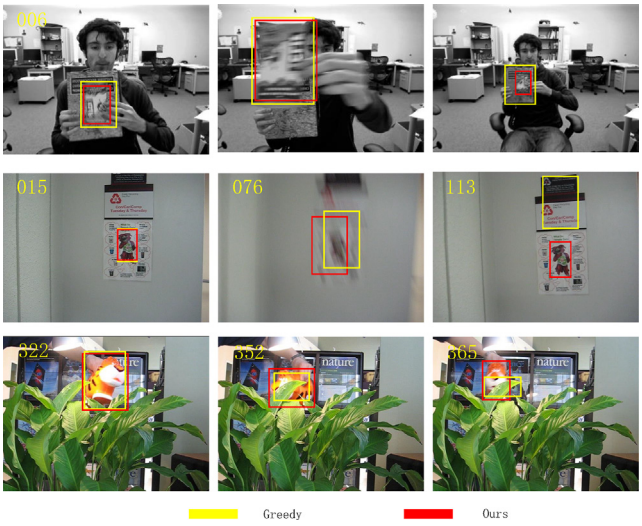


Fig. 7. Performance evaluation using greedy and our adaptive exploration method in *ClifBar*, *BlurOwl* and *Tiger2* video sequences. Yellow and red bounding boxes denote greedy and our adaptive exploration, respectively.

Table 5
Comparison with the three action choose methods on the OTB-100 dataset.

	I	II	III
Precision	0.880	0.891	0.897
AUC	0.629	0.637	0.642

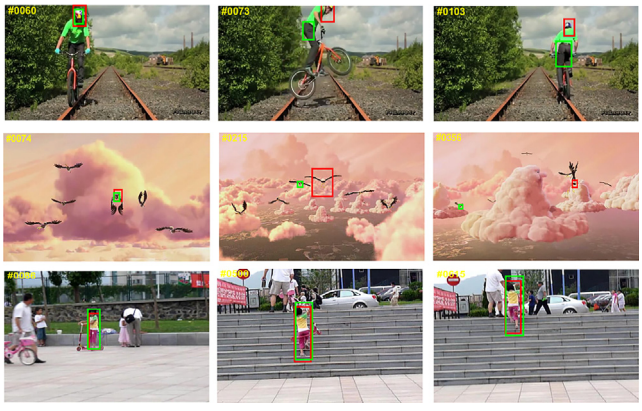


Fig. 8. Performance evaluation using regression and classification methods in *Biker*, *Bird1* and *Girl2* video sequences. Red and green bounding boxes denote regression and classification methods, respectively.

4.5. Evaluation on NFS, VOT, TC128, and UAV123 datasets

In order to further analyze the effectiveness of the proposed method, we choose four famous datasets. For each dataset, we select representative state-of-the-art tracking methods to compare with the proposed method, respectively.

We choose VOT16, and VOT18 to analyze the proposed method. VOT16 [16] contains 60 sequences, and all the trackers are evaluated by EAO (Expected Average Overlap), A (average overlap over successfully tracked frames), and R (failure rate). VOT18 adds more complex tracking sequences testing and uses the same evaluation method as VOT16. Our method performs better than CREST, which is RL-based. Because CREST has limit exploration ability, which shows that adaptive exploration for action space is crucial for robust tracking. SiamFC and SA-Siam ignore background information resulting in low robustness. The results in Table 7 shows the efficiency of the proposed method with an EAO score of 0.342 and achieves the best A and R scores among those trackers. Moreover, the proposed method also performs favorably against other methods in VOT18 dataset (see Table 8).

TC128 [15] contains 128 challenging colorful tracking sequences. The same evaluation setting is developed for TC128 and other datasets. 4 state-of-the-art methods (ECO, ADNet, DRL-IS [76], and SiamRPN [77]) are compared with our proposed method. ECO achieves a precision rate of 85.2% and our proposed method outperforms ADNet with an improvement of 3.8% shown in Table 9.

UAV123 [13] includes 123 tracking sequences and most of the are vehicles, which is harder to track while facing occlusion and out of view problem. Our method is compared with 12 state-of-the-art methods, the obtained precision are shown in Table 9. Our tracker outperforms ECO and achieves precision of 75.2%.

NFS [14] includes 100 sequences with 240 fps high frame rate. Most of those sequences are more than 5000 frames, we evaluate on the 240 fps version of the dataset with 8 state-of-the-art trackers. Table 10 shows success plot over 100 videos, reporting AUC scores in the legend. The proposed method significantly outperforms CCOT with a relative improvement of 4%.

5. Conclusion

In this paper, we propose a novel RL-based tracking method AEVRNet with non-convex optimized SVRG and adaptive exploration strategy. Firstly, the adaptive exploration strategy is proposed to combined temporal and spatial relations to expand action space and enhance exploration to escape local optima for object tracking. Secondly, SVRG backward propagation is presented to optimize supervised learning and RL for object tracking, which results in good convergence accuracy and speed. In particular, they are non-convex optimized, thus premature convergence to local optima is avoided. Thirdly, an action-reward loss function

Table 6
The results of reinforcement learning based tracker ACT with and without the proposed regression based action-reward loss on OTB-2013 and OTB-100.

	OTB-100		OTB-2013	
	Precision	AUC	Precision	AUC
ACT	0.859	0.648	0.884	0.667
ACT+regression	0.871	0.658	0.897	0.679

Table 7
Comparison with the state-of-the-art trackers on the VOT 2016 dataset. The results are presented in terms of expected average overlap (EAO), accuracy value (A), and robustness value (R). The best scores are highlighted in bold.

	Ours	CCOT	MDNet	SiamFC [30]	CREST	DSLT [68]	SA-Siam [33]	VITAL	Meta-Tracker [69]	RTINet [70]
EAO	**0.342**	0.331	0.257	0.277	0.283	0.332	0.291	0.323	0.314	0.298
R	**0.8**	0.85	1.204	1.382	1.083	0.93	1.08	0.97	0.934	1.07
A	**0.52**	0.523	0.533	0.549	0.524	0.525	0.54	0.531	0.521	0.57

Table 8
Comparison with the state-of-the-art trackers on the VOT 2018 dataset. The results are presented in terms of expected average overlap (EAO), accuracy value (A), and robustness value (R). The best scores are highlighted in bold.

	Ours	PG-Net	DRT [71]	MAML [72]	Siam R-CNN	SiamMask [73]	SiamRPN++ [74]	ATOM [75]	ECO	CCOT
EAO	**0.466**	0.447	0.356	0.392	0.408	0.347	0.414	0.401	0.280	0.267
R	**0.182**	0.192	0.201	0.22	0.22	0.288	0.234	0.204	0.276	0.318
A	**0.641**	0.618	0.519	0.635	0.609	0.602	0.6	0.59	0.484	0.494

Table 9
Precision and success plots of our proposed method on TC128 and UAV123. The best scores are highlighted in bold.

		Ours	ECO	ADNet	SiamRPN	DRL-IS
TC128	AUC	0.603	**0.605**	0.574	0.578	0.599
	Precision	0.821	**0.825**	0.783	0.799	0.818
UAV123	AUC	**0.531**	0.525	0.502	0.527	–
	Precision	**0.752**	0.741	0.716	0.748	–

Table 10
Comparison with the state-of-the-art trackers on the NFS dataset. The results are presented in terms of AUC. The best scores are highlighted in bold.

	Ours	ECO	Siamcar [78]	ADNet	CCOT [79]	Bridge [80]	DeepSRDCF	DaSiamRPN [81]	MDNet	SiamDW [82]
AUC	**0.532**	0.470	0.507	0.461	0.492	0.515	0.353	0.395	0.425	0.502

is designed by regression for object tracking, which is more sensitive to aspects of the target states, e.g., the width and height of the target and can further improve the accuracy of the proposed AEVRNet. Extensive experiments on six benchmarks show that the proposed method is favorably against the state-of-the-art RL-based tracking methods. In future work, we will try to introduce the attention mechanism with deep feature channel to improve the efficiency of feature utilization and enhance the robustness of the model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Yuxiang Yang: Conceptualization, Methodology, Software, Validation. **Weiwei Xing:** Writing - original draft, Writing - review & editing, Supervision. **Dongdong Wang:** Methodology, Software, Visualization, Investigation. **Shunli Zhang:** Writing - original draft, Writing - review & editing. **Qi Yu:** Data curation, Visualization, Investigation. **Liqiang Wang:** Writing - original draft, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, Imagenet classification with deep convolutional neural networks, in: Adv. Neural Inform. Process. Syst., 2012, pp. 1097–1105.
[2] Simone Bianco, Claudio Cusano, Raimondo Schettini, Single and multiple illuminant estimation using convolutional neural networks, IEEE Trans. Image Process. 26 (9) (2017) 4347–4362.
[3] Yuanpei Liu, Junbo Yin, Yu. Dajiang, Sanyuan Zhao, Jianbing Shen, Multiple people tracking with articulation detection and stitching strategy, Neurocomputing 386 (2020) 18–29.
[4] Lituan Wang, Lei Zhang, Zhang Yi, Trajectory predictor by using recurrent neural networks in visual tracking, IEEE Trans. Cybern. 47 (10) (2017) 3172–3183.
[5] Yunlong Wang, Fei Liu, Kunbo Zhang, Guangqi Hou, Zhenan Sun, Tieniu Tan, Lfnet: A novel bidirectional recurrent convolutional neural network for light-field image super-resolution, IEEE Trans. Image Process. 27 (9) (2018) 4274–4286.

[6] Yibing Song, Chao Ma, Xiaohe Wu, Lijun Gong, Linchao Bao, Wangmeng Zuo, Chunhua Shen, Rynson WH Lau, Ming-Hsuan Yang, Vital: Visual tracking via adversarial learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8990–8999..

[7] Wenbing Huang, Lijie Fan, Mehrtash Harandi, Lin Ma, Huaping Liu, Wei Liu, Chuang Gan, Toward efficient action recognition: Principal backpropagation for training two-stream networks, IEEE Trans. Image Process. 28 (4) (2018) 1773–1782.

[8] Xulun Ye, Jieyu Zhao, Long Zhang, Lijun Guo, A nonparametric deep generative model for multimanifold clustering, IEEE Trans. Cybern. 49 (7) (2019) 2664–2677.

[9] Ran Tao, Efstratios Gavves, Arnold WM Smeulders, Siamese instance search for tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1420–1429.

[10] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, Jin Young Choi, Action-decision networks for visual tracking with deep reinforcement learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2017, pp. 1349–1358..

[11] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, Huchuan Lu, Real-time 'actor-critic' tracking, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 328–345..

[12] Wu. Yi, Jongwoo Lim, Ming-Hsuan Yang, Online object tracking: A benchmark, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2013, pp. 2411–2418.

[13] Matthias Mueller, Neil Smith, Bernard Ghanem, A benchmark and simulator for uav tracking, in: Proc. Eur. Conf. Comput. Vis., 2016, pp. 445–461.

[14] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, Simon Lucey, Need for speed: A benchmark for higher frame rate object tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Pattern Recognit., 2017, pp. 1125–1134.

[15] Pengpeng Liang, Erik Blasch, Haibin Ling, Encoding color information for visual tracking: Algorithms and benchmark, IEEE Trans. Image Process. 24 (12) (2015) 5630–5644.

[16] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, Luka Čehovin, A novel performance evaluation methodology for single-target trackers, IEEE Trans. Pattern Anal. Mach. Intell. 38 (11) (2016) 2137–2155.

[17] Xingping Dong, Jianbing Shen, Yu. Dajiang, Wenguan Wang, Jianhong Liu, Hu.a. Huang, Occlusion-aware real-time object tracking, IEEE Trans. Multim. 19 (4) (2016) 763–771.

[18] Bo Ma, Lianghua Huang, Jianbing Shen, Ling Shao, Ming-Hsuan Yang, Fatih Porikli, Visual tracking under motion blur, IEEE Trans. Image Process. 25 (12) (2016) 5867–5876.

[19] Jianbing Shen, Yu. Dajiang, Leyao Deng, Xingping Dong, Fast online tracking with detection refinement, IEEE Trans. Intell. Transp. Syst. 19 (1) (2017) 162–173.

[20] Kai Chen, Wenbing Tao, Once for all: a two-flow convolutional neural network for visual tracking, IEEE Trans. Cir. Syst. Vid. Tech. (2017).

[21] Jianbing Shen, Zhiyuan Liang, Jianhong Liu, Hanqiu Sun, Ling Shao, Dacheng Tao, Multiobject tracking by submodular optimization, IEEE Trans. Cybern. 49 (6) (2018) 1990–2001.

[22] Du. Yihan, Yan Yan, Si Chen, Yang Hua, Object-adaptive LSTM network for real-time visual tracking with adversarial data augmentation, Neurocomputing 384 (2020) 67–83.

[23] Bo Ma, Hu. Hongwei, Jianbing Shen, Yangbiao Liu, Ling Shao, Generalized pooling for robust object tracking, IEEE Trans. Image Process. 25 (9) (2016) 4199–4208.

[24] Xin Wang, Zhiqiang Hou, Yu. Wangsheng, Zefenfen Jin, Yufei Zha, Xianxiang Qin, Online scale adaptive visual tracking based on multilayer convolutional features, IEEE Trans. Cybern. 49 (1) (2019) 146–158.

[25] Yuankai Qi, Lei Qin, Shengping Zhang, Qingming Huang, Hongxun Yao, Robust visual tracking via scale-and-state-awareness, Neurocomputing 329 (2019) 75–85.

[26] Lianghua Huang, Bo Ma, Jianbing Shen, Hui He, Ling Shao, Fatih Porikli, Visual tracking by sampling in part space, IEEE Trans. Image Process. 26 (12) (2017) 5800–5810.

[27] Hu. Hongwei, Bo Ma, Jianbing Shen, Hanqiu Sun, Ling Shao, Fatih Porikli, Robust object tracking using manifold regularized convolutional neural networks, IEEE Trans. Multim. 21 (2) (2019) 510–521.

[28] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, Ming-Hsuan Yang, Crest: Convolutional residual learning for visual tracking, in: Proc. IEEE Int. Conf. Comput. Vis., IEEE, 2017, pp. 2574–2583..

[29] Zheng Zhu, Guan Huang, Wei Zou, Du. Dalong, Chang Huang, Uct: learning unified convolutional networks for real-time visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 1973–1982.

[30] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, Philip HS Torr, Fully-convolutional siamese networks for object tracking, in: Proc. Eur. Conf. Comput. Vis., 2016, pp. 850–865.

[31] Jianbing Shen, Xin Tang, Xingping Dong, Ling Shao, Visual object tracking by hierarchical attention siamese network, IEEE Trans. Cybern. 50 (7) (2019) 3068–3080.

[32] Xingping Dong, Jianbing Shen, Wu. Dongming, Kan Guo, Xiaogang Jin, Fatih Porikli, Quadruplet network with one-shot learning for fast visual object tracking, IEEE Trans. Image Process. 28 (7) (2019) 3516–3527.

[33] Anfeng He, Chong Luo, Xinmei Tian, Wenjun Zeng, A twofold siamese network for real-time object tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4834–4843.

[34] Zhiyuan Liang, Jianbing Shen, Local semantic siamese networks for fast tracking, IEEE Trans. Image Process. 29 (2019) 3351–3364.

[35] Xingping Dong, Jianbing Shen, Triplet loss in siamese network for object tracking, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 459–474.

[36] Xiankai Lu, Chao Ma, Jianbing Shen, Xiaokang Yang, Ian Reid, Ming-Hsuan Yang, Deep object tracking with shrinkage loss, IEEE Trans. Pattern Anal. Mach. Intel., 2020..

[37] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, Stephen Maybank, Learning attentions: residual attentional siamese network for high performance online visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 4854–4863..

[38] Richard S. Sutton, Andrew G. Barto, et al., Reinforcement Learning: An Introduction, MIT press, 1998.

[39] Xingping Dong, Jianbing Shen, Wenguan Wang, Yu Liu, Ling Shao, Fatih Porikli, Hyperparameter optimization for tracking with continuous deep q-learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 518–527..

[40] Xingping Dong, Jianbing Shen, Wenguan Wang, Ling Shao, Haibin Ling, Fatih Porikli, Dynamical hyperparameter optimization via deep reinforcement learning in tracking, IEEE Trans. Pattern Anal. Mach. Intel. (2019).

[41] Chen Huang, Simon Lucey, Deva Ramanan, Learning policies for adaptive tracking with deep feature cascades, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 105–114.

[42] James Steven Supancic III, Deva Ramanan, Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 322–331.

[43] None, Visual tracking: An experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2013) 1442–1468..

[44] Léon Bottou, Large-scale machine learning with stochastic gradient descent, in: Proc. Int. Conf. Comput. Stat., 2010, pp. 177–186.

[45] Rie Johnson, Tong Zhang, Accelerating stochastic gradient descent using predictive variance reduction, in: Adv. Neural Inform. Process. Syst., 2013, pp. 315–323.

[46] Oron Anschel, Nir Baram, Nahum Shimkin, Averaged-dqn: Variance reduction and stabilization for deep reinforcement learning, in: Proc. Int. Conf. on Mach. Learn., 2017, pp. 176–185..

[47] Wu. Dongming, Xingping Dong, Jianbing Shen, Steven CH Hoi, Reducing estimation bias via triplet-average deep deterministic policy gradient, IEEE Trans. Neural Networks Learn. Syst. 31 (11) (2020) 4933–4945.

[48] Aloïs Pourchot, Olivier Sigaud, CEM-RL: combining evolutionary and gradient-based methods for policy search, in: Proc. Int. Conf.on Learn. Represent. OpenReview.net, 2019.

[49] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, Marcello Restelli, Stochastic variance-reduced policy gradient, in: Proc. Int. Conf. Mach. Learn., 2018.

[50] Branislav Kveton, Zheng Wen, Azin Ashkan, Csaba Szepesvari, Tight regret bounds for stochastic combinatorial semi-bandits, in: Artificial Intelligence and Statistics, 2015, pp. 535–543..

[51] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, Mubarak Shah, Visual tracking: An experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.

[52] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, Eco: Efficient convolution operators for tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6931–6939.

[53] Hyeonseob Nam, Bohyung Han, Learning multi-domain convolutional neural networks for visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4293–4302.

[54] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, Michael Felsberg, Convolutional features for correlation filter based visual tracking, in: Proc. IEEE Int. Conf. Comput. Vis. Workshops, 2015, pp. 58–66.

[55] Chao Ma, Jia-Bin Huang, Xiaokang Yang, Ming-Hsuan Yang, Hierarchical convolutional features for visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3074–3082.

[56] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, Ming-Hsuan Yang, Hedged deep tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4303–4311..

[57] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, Michael Felsberg, Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1430–1438.

[58] Jianming Zhang, Shugao Ma, Stan Sclaroff, Meem: robust tracking via multiple experts using entropy minimization, in: Proc. Eur. Conf. Comput. Vis., 2014, pp. 188–203.

[59] João F Henriques, Rui Caseiro, Pedro Martins, Jorge Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.

[60] Bingyan Liao, Chenye Wang, Yayun Wang, Yaonong Wang, Jun Yin, Pg-net: Pixel to global matching network for visual tracking, in: Proc. Eur. Conf. on Comput. Vis., 2020, pp. 429–444.

[61] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, Bastian Leibe, Siam r-cnn: Visual tracking by re-detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6578–6588.

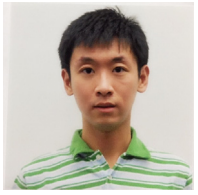
[62] Peixia Li, Boyu Chen, Wanli Ouyang, Dong Wang, Xiaoyun Yang, Huchuan Lu, Gradnet: Gradient-guided network for visual object tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6162–6171..

[63] Goutam Bhat, Martin Danelljan, Luc Van Gool, Radu Timofte, Learning discriminative model prediction for tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6182–6191.

[64] Xin Li, Chao Ma, Baoyuan Wu, Zhenyu He, Ming-Hsuan Yang, Target-aware deep tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1369–1378..
[65] Heng Fan, Haibin Ling, Siamese cascaded region proposal networks for real-time visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7952–7961.
[66] Pu. Shi, Yibing Song, Chao Ma, Honggang Zhang, Ming-Hsuan Yang, Deep attentive tracking via reciprocative learning, in: Proc. Int. Conf. on Neural Inform. Process. Syst., 2018, pp. 1935–1945.
[67] Guangting Wang, Chong Luo, Zhiwei Xiong, Wenjun Zeng, Spm-tracker: Series-parallel matching for real-time visual object tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3643–3652.
[68] Xiankai Lu, Chao Ma, Bingbing Ni, Xiaokang Yang, Ian Reid, Ming-Hsuan Yang, Deep regression tracking with shrinkage loss, in: Proc. Eur. Conf. Comput. Vis., 2011, pp. 353–369..
[69] Eunbyung Park, Alexander C Berg, Meta-tracker: Fast and robust online adaptation for visual object trackers, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 569–585.
[70] Yingjie Yao, Wu. Xiaohe, Lei Zhang, Shiguang Shan, Wangmeng Zuo, Joint representation and truncated inference learning for correlation filter based tracking, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 552–567.
[71] Chong Sun, Dong Wang, Huchuan Lu, Ming-Hsuan Yang, Correlation tracking via joint discrimination and reliability learning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 489–497..
[72] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, Wenjun Zeng, Tracking by instance detection: A meta-learning approach, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6288–6297.
[73] Qiang Wang, Li Zhang, Luca Bertinetto, Hu. Weiming, Philip HS Torr, Fast online object tracking and segmentation: A unifying approach, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1328–1338.
[74] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, Junjie Yan, Siamrpn++: Evolution of siamese visual tracking with very deep networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4282–4291..
[75] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, Michael Felsberg, Atom: Accurate tracking by overlap maximization, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4660–4669..
[76] Liangliang Ren, Xin Yuan, Lu. Jiwen, Ming Yang, and Jie Zhou. Deep reinforcement learning with iterative shift for visual tracking, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 697–713.
[77] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, Xiaolin Hu, High performance visual tracking with siamese region proposal network, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8971–8980..
[78] Dongyan Guo, Jun Wang, Ying Cui, Zhenhua Wang, Shengyong Chen, Siamcar: Siamese fully convolutional classification and regression for visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6269–6277.
[79] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, Michael Felsberg, Beyond correlation filters: Learning continuous convolution operators for visual tracking, in: Proc. Eur. Conf. Comput. Vis., 2016, pp. 472–488.
[80] Lianghua Huang, Xin Zhao, Kaiqi Huang, Bridging the gap between detection and tracking: A unified approach, in: Proc. IEEE Int. Conf. on Comput. Vis., 2019, pp. 3999–4009.
[81] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, Weiming Hu, Distractor-aware siamese networks for visual object tracking, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 101–117..
[82] Zhipeng Zhang, Houwen Peng, Deeper and wider siamese networks for real-time visual tracking, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4591–4600.

Yuxiang Yang received a BS degree in computer science and technology from Northeastern University of China, Liaoning, China, in 2014. He is currently a PhD student with the School of Software Engineering, Beijing Jiaotong University. His research interests include image processing, deep learning, reinforcement learning, and target tracking.

Weiwei Xing received a BS degree in computer science and technology and a PhD in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2001 and 2006, respectively. She is currently a Professor with the School of Software Engineering, Beijing Jiaotong University. Her research interests include intelligent information processing and machine learning.



Dongdong Wang received the Master degree in Environmental Science from Duke University in 2017 and is currently a Ph.D. student in School of Computer Science at University of Central Florida. His research interests include deep learning, computer vision, and distributed computing.

Shunli Zhang received BS and MS degrees in electronics and information engineering from Shandong University, Jinan, China, in 2008 and 2011, respectively, and a PhD in signal and information processing from Tsinghua University in 2016. He is currently a faculty member at the School of Software Engineering, Beijing Jiaotong University. His research interests include pattern recognition, computer vision, and image processing.

Qi Yu received a BS degree in software engineering from the School of Software, North University of China, Shanxi, China, in 2016. She is currently a master student with the School of Software Engineering, Beijing Jiaotong University. Her research interests include data analysis, machine learning, testing, and user experience.

Liqiang Wang received a PhD in computer science from Stony Brook University in 2006. He is an associate professor in the School of Computer Science at the University of Central Florida. His research interests include big data systems and deep learning. He received a US National Science Foundation CAREER Award in 2011.