Tracking-by-Fusion via Gaussian Process Regression Extended to Transfer Learning

Jin Gao, Qiang Wang, Junliang Xing, *Member, IEEE,* Haibin Ling, *Member, IEEE,*
Weiming Hu, *Senior Member, IEEE,* Stephen Maybank, *Fellow, IEEE*

Abstract—This paper presents a new Gaussian Processes (GPs)-based particle filter tracking framework. The framework non-trivially extends Gaussian process regression (GPR) to transfer learning, and, following the tracking-by-fusion strategy, integrates closely two tracking components, namely a GPs component and a CFs one. First, the GPs component analyzes and models the probability distribution of the object appearance by exploiting GPs. It categorizes the labeled samples into auxiliary and target ones, and explores unlabeled samples in transfer learning. The GPs component thus captures rich appearance information over object samples across time. On the other hand, to sample an initial particle set in regions of high likelihood through the direct simulation method in particle filtering, the powerful yet efficient correlation filters (CFs) are integrated, leading to the CFs component. In fact, the CFs component not only boosts the sampling quality, but also benefits from the GPs component, which provides re-weighted knowledge as latent variables for determining the impact of each correlation filter template from the auxiliary samples. In this way, the transfer learning based fusion enables effective interactions between the two components. Superior performance on four object tracking benchmarks (OTB-2015, Temple-Color, and VOT2015/2016), and in comparison with baselines and recent state-of-the-art trackers, has demonstrated clearly the effectiveness of the proposed framework.

Index Terms—Visual tracking, Gaussian processes, correlation filters, transfer learning, tracking-by-fusion.

1 Introduction

UNDERSTANDING how objects of interest move through video is one of the most fundamental problems in computer vision, as it can facilitate content-based semantic analysis for better video retrieval, real time human-computer interaction for more efficient computer understanding of human environments, object re-identification for multi-camera tracking in automated video surveillance, and registration or correct alignment of the virtual world with the real one for augmented reality, to name a few applications. There has been significant progress on accurate object detection and segmentation of interest over the recent years due largely to the use of convolutional neural networks [1], [2]. Online, robust tracking of the detected objects in video is required.

Given a detected object of interest, it is tempting to focus on distinguishing between the object and its neighboring background in subsequent frames. Meanwhile, adaptively updating the object observation model on the fly using the labeled samples obtained while tracking is widely adopted in many discriminative classifier based trackers [3], [4], [5], [6], [7], [8], [9], [10]. However, many of them [3], [4], [5], [6], [10] using particle filters simply estimate the probability distribution of object appearance by making its logistic transformation equivalent to the confidence of the classifier outputs. The optimization inconsistency resulting from the gap between maximizing the margin for classification (online labeled samples) and maximizing the conditional observation probability density of object appearance for tracking (unlabeled samples) is mostly ignored.

The past four years have witnessed an expeditious development in online visual tracking with several benchmarks proposed, e.g., OTB [11], [12], Temple-Color [13] and VOT [14], [15], etc. The soundness and fairness of these evaluation systems attract increasing attention from researchers in the tracking field. This also gives rise to many excellent tracking methods [9], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] in which the aforementioned inconsistency is reduced or avoided. The new methods include structured learning, multi-expert strategy, correlation filters (CFs) and deep learning. For example, the structured bounding box output in [9], the multiple instance partial-label learning setting for multi-expert tracking in [16], the ridge-regression based CFs in [17], [18], [19], [20], [21], [22], the attached bounding box regression in [24], the cross correlation in [25] and the saliency-map-based generative model in [23] preceded by a CNN are all dedicated to preventing the inconsistency between classification and tracking. Despite the lack of an explicit definition for the probability distribution of object appearance, all these inconsistency-preventing trackers compensate for the gap between classification and tracking adequately.

This paper has a new starting point for addressing the inconsistency issue in the traditional sequential Bayesian inference based particle filtering tracking framework [26], [27]. Specifically, inspired by GP classification from regression [28], [29], [30], GP regression [29] is extended to re-formulate the objective of the observation model in terms of transfer learning. Thus an approximation is obtained to the observation probability distribution directly from the GP model learning procedure, which is in contrast different from the logistic transformation with a separately learnt classifier. In this new approach to tracking, the online labeled samples collected after

- *J. Gao, Q. Wang, J. Xing and W. Hu are with the CAS Center for Excellence in Brain Science and Intelligence Technology, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China.*
  E-mail: {jin.gao, qiang.wang, jlxing, wmhu}@nlpr.ia.ac.cn
- *H. Ling is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA.*
  E-mail: hbling@temple.edu
- *S. Maybank is with the Department of Computer Science and Information Systems, Birkbeck College, Malet Street WC1E 7HX, London, UK.*
  E-mail: sjmaybank@dcs.bbk.ac.uk

tracking in the previous frames and the unlabeled samples that are tracking candidates corresponding to the particles in the current frame are fully exploited using GPs. An analytically tractable solution is achieved by introducing continuous latent variables for the unlabeled samples. These variables assist in predicting the tracking candidates' labels. On considering the different distortions of the object appearance over time (e.g., intrinsic object scale and pose variations, and variations caused by extrinsic illumination change, camera motion, occlusions and background clutter), it is necessary to have a large and diverse training set for updating the object observation model.

However, not all the labeled samples from the previous frames fit the current tracking task; in addition, temporary tracking failure, occlusions and potential sample misalignment in the previous frames can degrade the observation model update in the current frame. Therefore, in our transfer learning based new formulation, the labeled samples are divided into two categories, namely the auxiliary domain and the target domain. The auxiliary domain consists of samples from much earlier frames (auxiliary frames). The auxiliary samples cover the object appearance diversity. The target domain consists of samples from the most recent frames (target frames). These target samples are very closely related to the current tracking task. Continuous latent variables are introduced again for the auxiliary samples in each auxiliary frame and connected to the observed labels of themselves. This connection is based on a sigmoid noise output model so that the latent variables here can be thought of as the indicators for evaluating the extent to which the auxiliary samples in each auxiliary frame are related to the current tracking task. The indicators of the positive auxiliary samples also re-weight the relevance of the auxiliary frames to the current tracking task. In other words, our formulation can evaluate each auxiliary frame to see if it is relevant to the current tracking. The more closely the auxiliary frame is related to the current tracking task, the more important the role it may play in fitting the current tracking.

This new formulation is semi-supervised. The unlabeled samples contribute to the prior of GPs. The distribution of unlabeled samples influences both the re-weighting of the auxiliary frames and the final prediction of the tracking candidates' labels. So it is very important to generate the unlabeled samples properly according to a correct distribution. Encouraged by the most recently successful CFs-based tracking methods, we propose to use the response maps generated by the CFs as an approximation to the correct distribution for generating the corresponding particles. To this end, the rejection sampling based direct simulation method [31], [32] is used. The CF response maps enable us to evaluate the likelihood values in the rejection sampling process more efficiently as the values are only associated with each particle's location and scale in the current frame. This process encourages the particles to be in the right place both for our GP model learning and the final particle filters based tracking with the current observation incorporated. This is superior to the traditional particle filtering tracking methods which only use the prior transition probability for particle generation without incorporating the current observation.

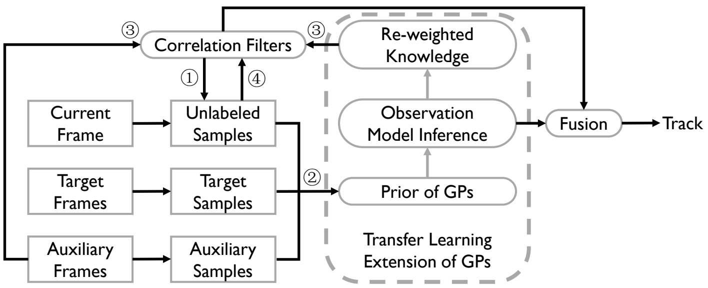

Fig. 1. The overview and system flowchart of the proposed formulation. The iteration loop labeled as steps ①, ② and ③ shows the interactions between the CFs and the transfer learning extension of GPs.

As [19] demonstrates that down-weighting the corrupted samples while increasing the weights of the correct ones improves the robustness of CFs in the SRDCF work [18], we exploit the re-weighting of the auxiliary frames for updating CFs. The re-weighted knowledge learnt from the auxiliary domain is exploited to generate the response maps in the current frame. The unlabeled samples corresponding to the particles generated from the current response maps influence the next re-weighting of the auxiliary frames. This iteration loop (see Fig. 1) generates unlabeled samples (step ①) both for re-weighting the auxiliary frames and inferencing GPs-based tracking task solution with transfer learning extension in the GPs component (step ②). Finally, we use the updated CFs (step ③) based on the re-weighted knowledge in the current frame to re-evaluate those unlabeled samples (step ④) and achieve an auxiliary CFs-based tracking task solution in the CFs component. This solution is fused with the GPs-based to make the final decision, leading to the transfer learning based fusion. Note that CFs are integrated naturally, and the interactions with GPs (steps ①, ② and ③) play an important role to achieve state-of-the-art performance. Superior results were achieved by integrating the CFs-based SRDCF tracker [18].

2 Related work

2.1 Tracking-by-Fusion

Tracking-by-fusion has proved valuable in numerous recent trackers. An advantage of this hybrid multi-expert strategy is that the information or knowledge from different sources can be used to model different distortions of the object appearance. Feature combination, expert ensemble and expert collaboration are three major paradigms in the tracking-by-fusion literature. Each paradigm can reduce the drift resulting from the direct Maximum a Posterior (MAP) estimation using a single expert.

Some local/global feature combination methods are described in [8], [33], [34]. Kwon et al. [35] integrate several decorrelated generative tracking models with different features in an interactive Markov Chain Monte Carlo (IMCMC) framework. The authors improved this original work using different estimation criteria [36], [37]. Some expert-ensemble-based methods also achieve impressive tracking performance. The base experts in each method are combined by a boosting algorithm [3], [4], or bootstrapped by structural constraints based P-N learning [38], or assigned different weights via randomized weighting vectors from a non-stationary distribution [39], or constructed with different features [40], or collected temporally from previous snapshots [16], or based on the hierarchical convolutional features from different CNN layers [41], [42]. Note that the base experts here for each method are always homogeneous. In contrast, there are many works which employ diverse tracking experts. These experts play complementary roles and offer different viewpoints. By using these experts that are biased in opposite directions and considering their results as alternatives, the true tracking result can be bracketed. This expert collaboration paradigm includes some generative-discriminative methods [8], [43], [44], [45],

[46], re-detection based methods [47], [48], [49], and methods combining template and colour-based models [50]. Our proposed tracking-by-fusion strategy bears some similarity to the expert collaboration paradigm, whereas the CFs and GPs based experts in our formulation are included in one loop (See Fig. 1) and hence interact more closely with each other, leading to mutual enhancement. This differs from the traditional expert collaboration methods, in which the experts have few interactions. It is noted that the novel joint learning framework for generative-discriminative experts based low-rank tracking methods proposed in [44], [45], [46] also have the advantage of expert mutual enhancement.

2.2 Transfer learning

There have been many efforts to use deep neural networks (DNNs) for transfer learning in online visual tracking. The first application of a DNN to robust visual tracking was made by Wang et al. [51] and based on unsupervised transfer learning. They exploit a stacked denoising autoencoder (SDAE) for unsupervised pre-training and representation learning from auxiliary data and then transfer the learnt features to the online tracking task. Since the hierarchical features learnt from convolutional neural networks (CNNs) outperform handcrafted features in numerous visual tasks such as image classification, recognition and detection on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), some supervised inductive transfer learning based CNN trackers [23], [24], [52] have been proposed. They pre-train (offline) networks using other tracking benchmarks or ILSVRC as source tasks and datasets, and then adapt the networks online to achieve high performance in the object visual tracking task. A clear deficiency in these approaches is the high computational demand. Contrary to these approaches, some supervised transfer learning based trackers using CNNs, but without fine-tuning, have been recently proposed to exploit the features from different hierarchical convolutional layers, to achieve comparable state-of-the-art results [20], [25], [41], [42], [53]. These three approaches all transfer the prior knowledge from offline training on the source tasks to the current online object tracking task. In contrast, we use prior knowledge extracted from the online GPs learning with the auxiliary samples.

Li et al. [4] first proposed to extend the semi-supervised on-line boosting tracker [3] using "Covariate Shift", leading to a novel semi-supervised transfer learning based tracker. It can utilize auxiliary and unlabeled samples effectively to train the strong classifier for tracking. The auxiliary samples' re-weighting in [4] is based on an online boosting classifier.

2.3 Correlation filtering

Recently, it has become increasingly popular to track by detecting patterns in image sequences by correlation with some example templates [17], [54], [55], [56]. Initially, Bolme et al. [54] develop a Minimum Output Sum of Squared Error (MOSSE) filter, and use a single feature channel, typically an intensity feature channel, for tracking. The success of the MOSSE tracker motivates a subsequent seminal work of Henriques et al. [17], [55], which presents a more robust kernelized CFs-based tracker (KCF) using a novel circular correlation solution in the Fourier domain for ridge regression based tracking. Its flexibility in incorporating multi-dimensional feature channels (ie., HOGs) and non-linear kernels produces a remarkable advance in performance, despite its high computational efficiency. There are four examples of research topics on CFs-based tracking dedicated to improving MOSSE and KCF.

First, some work adapts the CFs-based tracking framework to incorporate scale [56], [57] and rotation [58] estimates. Second, many researchers focus on making conceptual improvements in filter learning, such as alleviating the periodic effects of circular correlation [18], and modifying the filter update strategy to decrease the impact of corrupted training samples resulting from occlusions, misalignments and other perturbations [19]. Third, some trackers cast CFs as the base experts for ensemble [41], [42] or combine them with some instance-level object detectors [48], [49] to build multi-expert collaboration trackers. Finally, in addition to the incorporation of multi-dimensional feature channels, such as HOGs [17] and Color Names [59], many CFs-based trackers [20], [53], [60] exploit the CNN feature hierarchies for visual tracking. Our method is similar in spirit to the second and third topics in that we also integrate CFs into our formulation, and the CFs are updated using the learnt re-weighted knowledge. However, the ingredients (i.e., CFs and GPs) in our formulation have an impact on each other because they interact in a single loop (See Fig. 1). This is very different from the prior work [48], [49].

2.4 Subsequent work and Contributions

A preliminary version of this work, namely TGPR, was presented earlier [61]. TGPR analysed for the first time the probability distribution of object appearance in the Bayesian tracking framework using GPs, contrary to the common discriminative tracking approaches which by default formulate this probability distribution by making its logistic transformation equivalent to the confidence of the classifier outputs. The nontrivial extension of GPR to the transfer learning based tracking formulation equips TGPR with two properties, auxiliary samples' re-weighting and multi-expert collaboration; they both substantially improve the tracking performance on the OTB-2013 benchmark [11]. TGPR outperformed the other trackers by a large margin.

Afterwards, some authors built robust trackers in the spirit of our initial work on TGPR. In [19], a unified learning formulation is proposed to jointly optimize the object observation model parameters and the sample quality weights. This resembles TGPR, especially in the learning of real-valued weights to decrease the impact of corrupted samples. In [62], two differentially fine-tuned CNNs, one tuned conservatively and the other aggressively, make their own tracking decisions individually and the final integrated estimation is simply the more confident one. This method bears some similarity to TGPR, in that decisions from two differentially updated tracking experts are fused in order to reduce drifting. In recent work [63], a diversified ensemble discriminative tracker is proposed, which also draws support from an auxiliary classifier to break the self-learning loop using the effect of long-term memory and avoid the tracking drift.

In the present work, we add some important improvements to TGPR. We note the importance of the distribution of unlabeled samples for optimizing the performance of the GPs-based expert and assume that the response maps generated by the CFs approximate to the correct distribution (Section 3.3). In other words, the greater the response value a region has, the higher the probability of generating an unlabeled sample from that region. The optimized GPs-based expert provides the knowledge required to revise the re-weighting of the samples from the auxiliary frames for the update of the CFs-based expert. At the end of Section 3.2.4, we give more insight into how the distribution of unlabeled samples influences the re-weighting of auxiliary samples. The newly updated CFs-based expert re-evaluates those unlabeled samples more accurately and is again utilized for generating unlabeled samples in the next frame. Thus, we are able to fuse this auxiliary tracking task solution based on CFs with the target tracking task solution based on GPs to make the final decision. It is noted that the update of our CFs-based expert using the re-weighted knowledge is very different from the recently proposed unified formulation for adaptive decontamination of the training set in [19] in two aspects. First, our re-weighting only concentrates on the much earlier auxiliary frames while the latter decontamination includes the whole training set with the most recently collected training samples. Second, our update of the CFs-based expert in the current frame also relies on the extracted unlabeled samples, whereas the update in [19] only relies on the extracted labeled samples.

In addition, by integrating our formulation with SRDCF, we extend the original experiments on the OTB-2013 and VOT2013 benchmarks to the recent popular OTB-2015, Temple-Color, and VOT2015/2016 benchmarks. More comprehensive experiments are performed using the evaluation criteria TRE and SRE. All the tracking results are compared with many recent impressive state-of-the-art trackers. The expected average overlap (EAO) graphs and scores on VOT2015/2016 are also considered. Some variants of the new tracker are added for ablation study. We demonstrate the important role of the interactions between the integrated CFs and GPs in our formulation by making a comparison with experiments that omit these interactions. More considerable new analyses and intuitive explanations for these results are also provided.

3 Our Transfer Learning Based Formulation

In the following, we first analyse the objective of the observation model in the particle filters based tracking framework. Then we re-formulate it as a new transfer learning based formulation and extend GPs to approximately obtain its probability distribution. This process involves three sets of samples for robust tracking: auxiliary samples, target samples, and unlabeled samples. Two latent variables are introduced: one for re-weighting the auxiliary samples, and the other for deciding which tracking candidates are the best in the GPs-based tracking task solution. After giving more insight into the influence of the distribution of the unlabeled samples on the re-weighting of the auxiliary frames and hence on the update of CFs, we furthermore show how the unlabeled samples are generated from the response maps of the learnt CFs. Finally, we present a high level tracking pipeline to integrate the above fundamental components and describe the fusion strategy in our transfer learning based tracking formulation.

3.1 Byesian Inference Formulation Using Particle Filters

As detailed in [26], [27], visual tracking can be cast as a Bayesian inference problem in the particle filters based tracking framework. Given a set of observations $\mathcal{I}_t$ of the object up to the t-th frame, the optimal state variable $\hat{\boldsymbol{\ell}}_t$, which describes the object center location and scale at time t, can be estimated using the true posterior state density $p\left(\boldsymbol{\ell}_t|\mathcal{I}_t\right)$ with respect to different criteria, such as the *minimum mean-square error* (MMSE) estimate with the conditional mean taken and the *maximum a posteriori* (MAP) estimate with the mode taken. The posterior density $p\left(\boldsymbol{\ell}_t|\mathcal{I}_t\right)$ can be inferred using Bayesian theorem recursively through two steps,

$$p\left(\boldsymbol{\ell}_t|\mathcal{I}_t\right) \propto p\left(\mathbf{X}_t|\boldsymbol{\ell}_t\right) p\left(\boldsymbol{\ell}_t|\mathcal{I}_{t-1}\right) \tag{1}$$

$$p\left(\boldsymbol{\ell}_t|\mathcal{I}_{t-1}\right) = \int p\left(\boldsymbol{\ell}_t|\boldsymbol{\ell}_{t-1}\right) p\left(\boldsymbol{\ell}_{t-1}|\mathcal{I}_{t-1}\right) d\boldsymbol{\ell}_{t-1} \tag{2}$$

where $\mathbf{X}_t$ denotes the observation in the t-th frame, or more specifically the image region enclosed by $\boldsymbol{\ell}_t$, Eq. (2) involves the prediction step, and Eq. (1) carries out the update step. A good likelihood function $p\left(\mathbf{X}_t|\boldsymbol{\ell}_t\right)$ (also called the observation model) in the update step should modify the prior density $p\left(\boldsymbol{\ell}_t|\mathcal{I}_{t-1}\right)$ to obtain the sharply peaked posterior density.

Typically, there are two types of particle filters based Monte Carlo approximation approaches to solve this recursion problem by generating recursively a set of n_U particles, $\{\boldsymbol{\ell}_t^i, i = 1, 2, \ldots, n_U\}$. The first type is based on the sequential importance sampling (SIS) algorithm, which involves recursive propagation of importance weights and support particles as each measurement is received sequentially. One of the variants of SIS is the sampling importance re-sampling (SIR) algorithm which uses the prior transition probability $p\left(\boldsymbol{\ell}_t|\boldsymbol{\ell}_{t-1}\right)$ as the proposal density without incorporating the current observation. Despite the fact that SIR is sensitive to outliers, it has the advantage that the importance weights are easily evaluated when set to the likelihood values and the proposal density is also easily sampled. Thus, the MMSE estimate [27] can be taken as

$$\hat{\boldsymbol{\ell}}_t^{\mathrm{MMSE}} \approx \sum_{i=1}^{n_U} \frac{p\left(\mathbf{X}_t|\boldsymbol{\ell}_t = \boldsymbol{\ell}_t^i\right)}{\sum_{j=1}^{n_U} p\left(\mathbf{X}_t|\boldsymbol{\ell}_t = \boldsymbol{\ell}_t^j\right)} \boldsymbol{\ell}_t^i \tag{3}$$

and the MAP estimate as

$$\hat{\boldsymbol{\ell}}_t^{\mathrm{MAP}} \approx \arg\max_{\boldsymbol{\ell}_t^i} p\left(\mathbf{X}_t|\boldsymbol{\ell}_t = \boldsymbol{\ell}_t^i\right) \ . \tag{4}$$

The second type is the method of direct simulation [31], [32] which is a slight improvement over SIR based on the rejection sampling algorithm. This method directly uses the likelihood function to reject any proposed particles if they lead to unlikely predicted distributions for the observed data. The accepted particles then lead to the final required posterior distribution.

For each particle $\boldsymbol{\ell}_t^i$, the image region $\mathbf{X}_t^i$ associated with it is the measurement used in the likelihood function. This results in an image patch sample set $\mathcal{X}_U = \{\mathbf{X}_t^i, i = 1, 2, \ldots, n_U\}$, also called the unlabeled sample set in this paper. We concentrate on building a good GPs-based observation model by exploiting all the samples including these unlabeled samples. So there is a high demand for generating the particles properly according to a correct distribution. That means we can not directly use this observation model to conduct the direct simulation method for tracking. The SIR algorithm also suffers the disadvantage of incorporating no current observation into the proposal density for generating particles. So, we propose to first introduce the CFs-based likelihood function to conduct the direct simulation method and draw the particles approximately satisfying the posterior density. Then the SIR algorithm is conducted for MAP estimate using both the target and auxiliary observation models based on GPs and the updated CFs respectively. Note that the generated particles have uniform weights for the attached SIR algorithm. Finally, our transfer learning based tracking-by-fusion formulation is detailed in Section 3.3. Below we detail the GPs-based observation model firstly.

3.2 GPs-Based Observation Model

We specify our GP observation model based on the tracking results $\{\hat{\boldsymbol{\ell}}_f, f = 1, 2, \ldots, t-1\}$ up to the $(t-1)$-st frame. We collect n_L training samples, each with an indicator belonging to $\{-1, +1\}$, from the previous $t-1$ frames and maintain them over time. We call them the labeled data. The indicator

"+1" means the labeled sample has the "same" observation to the object, and vice versa. Furthermore, we divide these training samples into two categories and treat them differently: the auxiliary samples are updated slowly and carefully; the target samples are updated quickly and aggressively. Let $\mathcal{D}_T = \{(\mathbf{X}^j, y_j), j = 1, 2, \ldots, n_T\}$ denote the target sample set, and $\mathcal{D}_A = \{(\mathbf{X}^j, y_j), j = n_T + 1, n_T + 2, \ldots, n_T + n_A\}$ the auxiliary sample set, where $n_L = n_T + n_A$ and $y_j \in \{-1, +1\}$ is the indicator variable. Then, our GPs-based observation model is specified as: for each particle $\boldsymbol{\ell}_t^i$, the measurement density value $p\left(\mathbf{X}_t | \boldsymbol{\ell}_t = \boldsymbol{\ell}_t^i\right)$ is proportional to the probability of the measurement $\mathbf{X}_t^i$ having the "same" observation to the object, i.e., $\Pr\left(y_i = +1 | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$, where y_i is the indicator for $\mathbf{X}_t^i$. We can also cast these conditional probabilities as a regression function of the unlabeled samples for the indicator variables.

As in our initial work TGPR [61], we pick out our favorite smoother and directly estimate this regression function for all y_i in $\mathbf{y}_U = \left[y_1, y_2, \ldots, y_{n_U}\right]^\top$. Denote the observed indicators of the target and auxiliary samples as $\mathbf{y}_T = \left[y_1, y_2, \ldots, y_{n_T}\right]^\top$ and $\mathbf{y}_A = \left[y_{n_T+1}, y_{n_T+2}, \ldots, y_{n_L}\right]^\top$. Let $\mathbf{1} = [+1, +1, \ldots, +1]^\top$, then the regression function for $\mathbf{y}_U$ can be written as

$$\mathcal{R} = \mathbf{Pr}\left(\mathbf{y}_U = \mathbf{1} | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) \tag{5}$$

where $\mathbf{Pr}\left(\mathbf{y}_U = \mathbf{1} | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) = [\Pr(y_1 = +1 | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T), \ldots, \Pr(y_{n_U} = +1 | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T)]^\top$. Inspired by GP classification from regression [28], [29], [30], we introduce two real-valued latent variables $\mathbf{z}_A = \left[z_{n_T+1}, z_{n_T+2}, \ldots, z_{n_L}\right]^\top \in \mathbb{R}^{n_A}$ and $\mathbf{z}_U = \left[z_1, z_2, \ldots, z_{n_U}\right]^\top \in \mathbb{R}^{n_U}$ corresponding to $\mathbf{y}_A$ and $\mathbf{y}_U$ respectively to analyse the regression $\mathcal{R}$ directly, and marginalize $\mathcal{R}$ over $\mathbf{z}_A$, $\mathbf{z}_U$ at $\mathcal{D}_A$ and $\mathcal{X}_U$,

$$\begin{aligned} &\mathbf{Pr}\left(\mathbf{y}_U = \mathbf{1} | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) \\ =&\mathbf{E}_{\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T}\left[\, \mathbf{Pr}\left(\mathbf{y}_U = \mathbf{1} | \mathbf{z}_A, \mathbf{z}_U, \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) \right] \\ =& \int\!\!\int \mathbf{Pr}\left(\mathbf{y}_U = \mathbf{1} | \mathbf{z}_U\right) p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)\, d\mathbf{z}_A\, d\mathbf{z}_U \end{aligned} \tag{6}$$

where $p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$ is a probability density function.

3.2.1 Label generation process.

As in [28], [29], [30], We also model $\mathbf{Pr}\left(\mathbf{y}_U | \mathbf{z}_U\right)$ as a noisy label generation process $\mathcal{X}_U \to \mathbf{z}_U \to \mathbf{y}_U$ with the sigmoid noise output model:

$$\Pr\left(y_i | z_i\right) = \frac{e^{\gamma z_i y_i}}{e^{\gamma z_i y_i} + e^{-\gamma z_i y_i}} = \frac{1}{1 + e^{-2\gamma z_i y_i}} \ , \tag{7}$$

$\forall i = 1, 2, \ldots, n_U$, where γ is a parameter controlling the steepness of the sigmoid. Intuitively, the larger $|z_i|$, the more likely that the candidate $\mathbf{X}_t^i$ has the indicator variable $y_i = \mathrm{sign}(z_i)$. This generation process is also transferable to the auxiliary data generation, i.e., $\mathcal{X}_A \to \mathbf{z}_A \to \mathbf{y}_A$, where $\mathcal{X}_A = \{\mathbf{X}^j, j = n_T + 1, n_T + 2, \ldots, n_T + n_A\}$. In this case, $\mathbf{z}_A$ can be thought as the re-weighted knowledge extracted from the regression $\mathcal{R}$ and is related with $\mathbf{y}_A$ via a sigmoid noise output model similar to Eq. (7). Thus, $\mathbf{z}_A$ plays a linking role between the regression of GPs-based tracking task solution in Section 3.2.2 and the indicators of the auxiliary samples. The replacement of $\mathbf{y}_A$ by $\mathbf{z}_A$ in the decision making of this solution reduces the impact of the corrupted auxiliary samples.

3.2.2 GPs extended to transfer learning.

We use Bayesian theorem to analyse the density in Eq. (6):

$$\begin{aligned} &p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) \\ =&\frac{\Pr\left(\mathbf{y}_A | \mathbf{z}_A, \mathbf{z}_U, \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\ p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)}{\Pr\left(\mathbf{y}_A | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)} \\ \propto&\Pr\left(\mathbf{y}_A | \mathbf{z}_A\right)\ p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right) \ . \end{aligned} \tag{8}$$

Note that the normalization term $\Pr\left(\mathbf{y}_A | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)$ is skipped without altering the analysis of Eq. (6) as this term can be taken out of the integrand. The term $p(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)$ is assumed to define a Gaussian stochastic process, which is a collection of random variables indexed by the samples in $\mathcal{X}_A$ and $\mathcal{X}_U$. It is specified by giving the expected value $\boldsymbol{\mu}$ and the $(n_A + n_U) \times (n_A + n_U)$ covariance matrix $\mathbf{G}$:

$$p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right) = \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{G}\right) \ . \tag{9}$$

We can determine $\boldsymbol{\mu}$ and $\mathbf{G}$ based on $\mathcal{X}_A$, $\mathcal{X}_U$ and $\mathcal{D}_T$.

Initially, we define a Gram matrix $\mathbf{G}_{\text{all}}$ (symmetric, positive-semidefinite) based on all samples (auxiliary, target and unlabeled), and it can be thought as the prior of GPs for our observation model inference. Typically, elements of a Gram matrix store the dot-products in a higher-dimensional space between all pairs of samples by transforming the original sample to that space. Without creating vectors in that space, we only need to evaluate dot-products using a kernel function. If we introduce an additional latent variable $\mathbf{z}_T = \left[z_{n_1}, z_{n_2}, \ldots, z_{n_T}\right]^\top \in \mathbb{R}^{n_T}$ corresponding to $\mathbf{y}_T$, Eq. (9) can be represented as a conditional probability density function

$$p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right) = \frac{p\left(\mathbf{z}_A, \mathbf{z}_U, \mathbf{z}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)}{p\left(\mathbf{z}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)} \ , \tag{10}$$

where the joint probability density function $p(\mathbf{z}_A, \mathbf{z}_U, \mathbf{z}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T)$ also defines a Gaussian stochastic process specified by $\mathcal{N}\left(\mathbf{0}, \mathbf{G}_{\text{all}}\right)$, and the marginal probability density function $p\left(\mathbf{z}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)$ has a constant value at a given $\mathbf{z}_T = \mathbf{y}_T$. Thus, we derive $\boldsymbol{\mu}$ and $\mathbf{G}$ as follows.

Proposition 1. Take the logarithm of $p(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T)$

$$\begin{aligned} &\ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\right) \\ &= -\frac{1}{2}\Big(\begin{pmatrix}\mathbf{y}_T^\top & \mathbf{z}_A^\top & \mathbf{z}_U^\top\end{pmatrix} \mathbf{G}_{\text{all}}^{-1} \begin{pmatrix}\mathbf{y}_T \\ \mathbf{z}_A \\ \mathbf{z}_U\end{pmatrix} \Big) + c \ , \end{aligned} \tag{11}$$

where c is a constant value. Let

$$\mathbf{G}_{\text{all}}^{-1} = \begin{pmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{M}\end{pmatrix} \ , \tag{12}$$

we determine $\boldsymbol{\mu}$ and $\mathbf{G}$ as: $\boldsymbol{\mu} = -\mathbf{M}^{-1}\mathbf{B}^\top\mathbf{y}_T$ and $\mathbf{G} = \mathbf{M}^{-1}$.

The derivation of this proposition is given in Appendix A.2.

From Eq. (8) we see that, the non-Gaussianity of $\Pr\left(\mathbf{y}_A | \mathbf{z}_A\right)$ makes the posterior $p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$ no longer Gaussian, consequently Eq. (6) becomes analytically intractable. According to [28], [29], [64], assuming $p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$ to be uni-modal, we can consider instead its *Laplace approximation*. In place of the correct density we use an $(n_A + n_U)$-dimensional Gaussian measure with expected value $\boldsymbol{\mu}' \in \mathbb{R}^{n_A + n_U}$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{(n_A+n_U)\times(n_A+n_U)}$, where

$$\boldsymbol{\mu}' = \mathop{\arg\max}_{\mathbf{z}_A \in \mathbb{R}^{n_A}, \mathbf{z}_U \in \mathbb{R}^{n_U}} p(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) \ . \tag{13}$$

We decompose this maximization over $\mathbf{z}_A$ and $\mathbf{z}_U$ separately.

Taking the logarithm of Eq. (8), we get the following objective function to maximize

$$\mathcal{J} = \underbrace{\ln\left(\Pr\left(\mathbf{y}_A | \mathbf{z}_A\right)\right)}_{Q_1(\mathbf{z}_A)} + \underbrace{\ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\right)}_{Q_2(\mathbf{z}_A, \mathbf{z}_U)} \ . \tag{14}$$

Note $\mathbf{z}_U$ only appears in Q_2, and we can independently maximize $Q_2(\mathbf{z}_A, \bullet)$ w.r.t. $\mathbf{z}_U$ given $\hat{\mathbf{z}}_A$, where $(\hat{\mathbf{z}}_A, \hat{\mathbf{z}}_U) = \arg\max_{\mathbf{z}_A, \mathbf{z}_U} Q_1 + Q_2$. Let

$$\mathbf{G}_{\text{all}} = \begin{pmatrix}\mathbf{G}_{LL} & \mathbf{G}_{LU} \\ \mathbf{G}_{UL} & \mathbf{G}_{UU}\end{pmatrix} \ . \tag{15}$$

According to [28], [64], by taking derivative of $Q_2(\mathbf{z}_A, \bullet)$ w.r.t. $\mathbf{z}_U$, the optimal value $\hat{\mathbf{z}}_U$ can be analytically derived as:

$$\hat{\mathbf{z}}_U = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_T \\ \hat{\mathbf{z}}_A\end{pmatrix} \quad . \tag{16}$$

Thus we can derive $\hat{\mathbf{z}}_A$ from Eq. (14) as follows.

Proposition 2. The optimal value $\hat{\mathbf{z}}_A$ is formally given by:

$$\begin{aligned}\hat{\mathbf{z}}_A = \underset{\mathbf{z}_A \in \mathbb{R}^{n_A}}{\arg\max} & \sum_{j=n_T+1}^{n_L} \ln\left(\Pr\left(y_i|z_i\right)\right) \\ & - \frac{1}{2}\begin{pmatrix}\mathbf{y}_T^\top & \mathbf{z}_A^\top\end{pmatrix}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_T \\ \mathbf{z}_A\end{pmatrix} + c \ .\end{aligned} \tag{17}$$

The derivation of this proposition is given in Appendix A.3.

The above derivations in Eqs. (16) and (17) help us to analytically compute $\boldsymbol{\mu}'$. In fact, we can also analytically compute the covariance $\boldsymbol{\Sigma}$ and thus Eq. (6) is computationally feasible. That is because determining Eq. (6) reduces to

$$\begin{aligned}&\Pr\left(\mathbf{y}_U = \mathbf{1}|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right) \\ &= \int_{\mathbb{R}^{n_U}} \Pr\left(\mathbf{y}_U = \mathbf{1}|\mathbf{z}_U\right) p(\mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T) d\mathbf{z}_U \ ,\end{aligned} \tag{18}$$

where $p\left(\mathbf{z}_U|\mathcal{X}_U, \mathcal{D}_A, \mathcal{D}_T\right)$ is approximatively a Gaussian density $\mathcal{N}\left(\hat{\mathbf{z}}_U, \boldsymbol{\Sigma}_{UU}\right)$, and $\boldsymbol{\Sigma}_{UU}$ is the bottom-right block of $\boldsymbol{\Sigma}$ (see [28] for more details). However, in practice, we only need to exploit the fact that the larger $\hat{z}_i$ in $\hat{\mathbf{z}}_U$, the more likely $\mathbf{X}_t^i$ has the "same" observation to the object ($y_i = +1$).

3.2.3 Iterative solution for the re-weighted knowledge.

We use an iterative Newton-Raphson update to find the optimal value $\hat{\mathbf{z}}_A$ in Proposition 2. Let $\rho(z_j) = 1/\left(1 + e^{-2\gamma z_j}\right)$, where $j = n_T+1, n_T+2, \ldots, n_T+n_A$. Because $y_j \in \{-1, +1\}$, the auxiliary data generation model can be written as

$$\Pr\left(y_j|z_j\right) = \frac{1}{1 + e^{-2\gamma z_j y_j}} = \rho(z_j)^{\frac{y_j+1}{2}}(1 - \rho(z_j))^{\frac{1-y_j}{2}} \ \ , \tag{19}$$

therefore

$$Q_1(\mathbf{z}_A) = \gamma\left(\mathbf{y}_A - \mathbf{1}\right)^\top \mathbf{z}_A - \sum_{j=n_T+1}^{n_L} \ln\left(1 + e^{-2\gamma z_j}\right) \ \ . \tag{20}$$

Let

$$\mathbf{G}_{LL}^{-1} = \begin{pmatrix}\mathbf{F}_{TT} & \mathbf{F}_{TA} \\ \mathbf{F}_{AT} & \mathbf{F}_{AA}\end{pmatrix} \ \ , \tag{21}$$

we can obtain $\hat{\mathbf{z}}_A$ by taking derivative of $Q_1 + Q_2$ w.r.t. $\mathbf{z}_A$,

$$\begin{aligned}\frac{\partial(Q_1 + Q_2)}{\partial \mathbf{z}_A} &= \gamma(\mathbf{y}_A - \mathbf{1}) + 2\gamma\left(\mathbf{1} - \boldsymbol{\rho}(\mathbf{z}_A)\right) \\ &\quad - \mathbf{F}_{AA}\mathbf{z}_A - \frac{1}{2}\mathbf{F}_{TA}^\top\mathbf{y}_T - \frac{1}{2}\mathbf{F}_{AT}\mathbf{y}_T \ \ ,\end{aligned} \tag{22}$$

where $\boldsymbol{\rho}(\mathbf{z}_A) = [\rho(z_{n_T+1}), \rho(z_{n_T+2}), \ldots, \rho(z_{n_L})]^\top$. The term $\boldsymbol{\rho}(\mathbf{z}_A)$ makes it impossible to compute the roots $\hat{\mathbf{z}}_A$ in a closed form. We instead solve it with the Newton-Raphson algorithm,

$$\mathbf{z}_A^{m+1} \leftarrow \mathbf{z}_A^m - \eta \cdot \mathbf{H}^{-1} \cdot \frac{\partial(Q_1 + Q_2)}{\partial \mathbf{z}_A}\Big|_{\mathbf{z}_A^m} \tag{23}$$

where $\eta \in \mathbb{R}^+$ is chosen such that $(Q_1+Q_2)^{m+1} > (Q_1+Q_2)^m$, and $\mathbf{H}$ is the Hessian matrix defined as

$$\mathbf{H} = \left[\frac{\partial^2(Q_1 + Q_2)}{\partial z_i \partial z_j}\Big|_{\mathbf{z}_A}\right] = -\mathbf{F}_{AA} - \mathbf{P} \tag{24}$$

where $\mathbf{P}$ is a diagonal matrix with $P_{ii} = 4\gamma^2\rho(z_i)(1 - \rho(z_i))$.

3.2.4 Construction of the Gram matrix.

A very important aspect of GPs for our observation model inference lies in constructing the prior Gram or kernel matrix $\mathbf{G}_{\text{all}}$ in Eq. (11). Some methods define the entries of such matrices in a "local" manner. For example, in a radial basis function (RBF) kernel matrix $\mathbf{K}$, the matrix entry $k_{ij} = \exp(-d_{ij}^2/\alpha^2)$ depends only on the distance d_{ij} between the i, j-th items. In this case unlabeled samples are useless for calculating $\hat{\mathbf{z}}_A$ because the influence of such samples in solving Eq. (17) is marginalized out. Addressing this issue, we instead define the Gram matrix $\mathbf{G}_{\text{all}}$ based on a weighted graph to explore the manifold structure of all samples (both labeled and unlabeled), as suggested in [64], [65], following the intuition that similar samples often share similar labels.

Consider a graph $\mathcal{G} = (V, E)$ with node set $V = T \cup A \cup U$ corresponding to all the $n = n_L+n_U$ samples, $T = \{1, \ldots, n_T\}$ the labeled target samples, $A = \{n_T + 1, \ldots, n_T + n_A\}$ the labeled auxiliary samples, and $U = \{n_L + 1, \ldots, n_L + n_U\}$ the unlabeled samples. We define an $n \times n$ symmetric weight matrix $\mathbf{W} = [w_{ij}]$ on the edges of the graph mimicking the local patch representation method in [66]. This benefits the robust tracking, especially under partial occlusion. For the i-th and j-th samples, the weight w_{ij} is defined by the spatially weighted Euclidean metric over the image representation, i.e., HOGs in particular Felzenszwalb's variant [67]. Specifically, for the i-th sample, we divide its image representation into $N_{\text{r}} \times N_{\text{c}}$ non-overlapping blocks, and then describe its (p, q)-th block using a feature vector $\mathbf{h}_i^{pq}$ obtained by concatenating the histogram orientation bins in that block. Thus, w_{ij} is defined as

$$w_{ij} = \frac{1}{\sum_{p,q}\beta_{p,q}}\sum_{p,q}\beta_{p,q}\exp\Big(-\frac{\|\mathbf{h}_i^{pq} - \mathbf{h}_j^{pq}\|^2}{\sigma_i^{pq}\sigma_j^{pq}}\Big) \tag{25}$$

where σ_i^{pq} is a local scaling factor proposed by [68]; $\beta_{p,q} = \exp(-\|\text{pos}^{pq} - \text{pos}^o\|^2/2\sigma_{\text{spatial}}^2)$ is the spatial weight, in which pos^{pq} indicates the position of the (p, q)-th block, pos^o the position of the block center, and σ_{spatial} the scaling factor.

Instead of connecting all the pairs of nodes in V, we restrict the edges to be within k-nearest-neighborhood, where large distance between two nodes corresponds to small edge weight between them. The parameter k controls the density of the graph and thus the sparsity of $\mathbf{W}$. We define the combinatorial Laplacian of $\mathcal{G}$ in the matrix form as $\boldsymbol{\Delta} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \text{diag}(D_{ii})$ is the diagonal matrix with $D_{ii} = \sum_j w_{ij}$.

Finally, we define the Gram matrix or kernel by $\mathbf{G}_{\text{all}} = (\boldsymbol{\Delta}+\mathbf{I}/\lambda^2)^{-1}$, where the regularization term $\mathbf{I}/\lambda^2$ guards $\boldsymbol{\Delta} + \mathbf{I}/\lambda^2$ from being singular. From the definition of $\mathbf{G}_{\text{all}}$ we can see that, the prior covariance in Eq. (11) between any two samples i, j in general depends on all entries in $\boldsymbol{\Delta}$ – the distances between all the pairs of the target and unlabeled samples are used to define the prior. Thus, distribution of target and unlabeled samples may strongly influence the kernel, which is desired both for extracting the re-weighted knowledge $\hat{\mathbf{z}}_A$ and solving the GPs-based tracking task solution in Eq. (16). We can trace back to Eq. (17) and see more discussion details about it below.

Discussion. In Eq. (17), the first term $Q_1(\mathbf{z}_A)$ in the objective function is to measure the consistencies between the elements of latent variable $\mathbf{z}_A$ and their corresponding observed indicators in $\mathbf{y}_A$ with the relationship modeled as the sigmoid noisy label generation process as in Eq. (7).

As for the second term in Eq. (17), we let $\mathbf{G}_{\text{all}} = \boldsymbol{\Delta}^{-1}$ without loss of generality to facilitate the analysis of how the distribution of unlabeled samples influences the re-weighting of auxiliary samples. Recall Eq. (15) and let

$$\mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_{LL} & \mathbf{\Delta}_{LU} \\ \mathbf{\Delta}_{UL} & \mathbf{\Delta}_{UU} \end{pmatrix} \ . \tag{26}$$

Using the partitioned matrix inversion theorem given in Eq. (35) of Appendix A.1, we can derive $\mathbf{G}_{LL}^{-1}$ in the second term of Eq. (17) as follows:

$$\mathbf{G}_{LL}^{-1} = \mathbf{\Delta}_{LL} - \mathbf{\Delta}_{LU}\mathbf{\Delta}_{UU}^{-1}\mathbf{\Delta}_{UL} \ . \tag{27}$$

Meanwhile, according to Eq. (34) in Appendix A.1, it is straightforward to have

$$\mathbf{\Delta}_{UU}^{-1}\mathbf{\Delta}_{UL} = \left(\mathbf{\Delta}_{LU}\mathbf{\Delta}_{UU}^{-1}\right)^{\top} = -\mathbf{G}_{UL}\mathbf{G}_{LL}^{-1} \ . \tag{28}$$

Denote $\mathbf{y} = \begin{pmatrix} \mathbf{y}_T \\ \mathbf{z}_A \end{pmatrix}$, then the second term of Eq. (17) can be decomposed into:

$$\begin{aligned} \mathbf{y}^{\top}\mathbf{G}_{LL}^{-1}\mathbf{y} =& \mathbf{y}^{\top}\mathbf{\Delta}_{LL}\mathbf{y} + \mathbf{y}^{\top}\mathbf{\Delta}_{LU}\mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y} \\ =& \mathbf{y}^{\top}\mathbf{\Delta}_{LL}\mathbf{y} + \mathbf{y}^{\top}\mathbf{\Delta}_{LU}\mathbf{z}_U \\ =& \mathbf{y}^{\top}\mathbf{\Delta}_{LL}\mathbf{y} + \mathbf{y}^{\top}\mathbf{\Delta}_{LU}\mathbf{z}_U + \mathbf{z}_U^{\top}\mathbf{\Delta}_{UL}\mathbf{y} + \mathbf{z}_U^{\top}\mathbf{\Delta}_{UU}\mathbf{z}_U \\ =& \frac{1}{2}\sum_{j,j^*=1,j\neq j^*}^{n_L} w_{jj^*}\left(z_j - z_{j^*}\right)^2 + \sum_{i=1}^{n_U}\sum_{j=1}^{n_L} w_{ij}\left(z_i - z_j\right)^2 \\ &+ \frac{1}{2}\sum_{i,i^*=1,i\neq i^*}^{n_U} w_{ii^*}\left(z_i - z_{i^*}\right)^2 \ , \end{aligned} \tag{29}$$

where $z_i = y_i$ for $i = 1, \ldots, n_T$.

From the above derivations, we can easily find that the second term in Eq. (17) measures the inconsistencies between the re-weighted knowledge z_i of auxiliary samples and the observed indicators y_i of target samples, and there is a high demand for the minimization of these inconsistencies. More specifically, we can interpret this minimization as follows. i) For each of the auxiliary samples directly linked to the target samples, z_i is forced to be similar to y_i of the nearby target samples. ii) For the auxiliary samples directly linked to each other, they will have similar z_i if they are nearby. iii) For the auxiliary samples linked to the target samples indirectly through the unlabeled samples, the unlabeled samples may force the nearby auxiliary samples to have z_i be similar to y_i of the nearby target samples. Until now we have given more insight into how the distribution of unlabeled samples influences the re-weighting of auxiliary samples. Moreover, our transfer learning based formulation can be interpreted as the learning of re-weighted knowledge for the auxiliary samples, which are mostly influenced by the target and unlabeled samples related to the current tracking task. The more closely the auxiliary samples are related to the current tracking task when the inconsistencies in Eq. (17) are minimized, the more important role they may play in fitting the current tracking task properly and the larger absolute weight they may be given.

3.3 Building Blocks Using Correlation Filters

Since our transfer learning based formulation is inferred in a semi-supervised fashion as detailed in Section 3.2.4, the question is how to generate such a suitable particle set to facilitate this formulation. The ideal manner is by setting the proposal density function to the true posterior density $p\left(\boldsymbol{\ell}_t|\mathcal{I}_t\right)$ itself, however, this is unrealistic. The SIR-based tracking methods generate the particles only by using $p\left(\boldsymbol{\ell}_t|\boldsymbol{\ell}_{t-1}\right)$ as the proposal density without the current observation incorporated, and hence tend to cause tracking drift.

Recently, CFs-based trackers (e.g., [17], [18], [20]) have obtained much attention in the literature. They approximate the true posterior state density as the generated response maps and obtain the state of the object with the maximum response value. This approximation is much better, but still not enough. The limitation is that CFs are updated by combining the new template computed using the new tracking result with the previous template only using a rolling average manner, which tends to miss the chance to fully exploit the relationships between the historic tracking results and the current frame. Based on these observations, we derive some inspiration from CFs and propose to cast the generated response maps in the current frame as an approximation to the correct density for generating the particles using the direct simulation method. The advantage is not only that the current observation is incorporated into the particle generation process, but also that this process relies on another complementary expert with a different viewpoint, which is beneficial for fusion.

In the next, we first give some preliminary knowledge about CFs in the tracking literature. To concretize our idea of generating particles, we then detail the procedure of sampling using the direct simulation method. Finally, we update CFs using the re-weighted knowledge learnt from our transfer learning based formulation and then re-evaluate the unlabeled samples using the updated CFs, leading to a tracking-by-fusion strategy for making the final decision.

3.3.1 *Preliminaries.*

We extend the depiction of the formulation of multi-channel CFs in [18] to a more general multi-channel multi-resolution case. Specifically, the correlation filter h is learnt from a set of training examples $\{(\mathbf{\Phi}(\hat{\boldsymbol{\ell}}_{s,f(a)}), y_s)\}_{s=1,a=1}^{S,|A|}$, where S denotes the number of resolutions, $f(\cdot)$ is the time index for the frames in the training set A of $|A|$ frames. Each training sample $\mathbf{\Phi}(\hat{\boldsymbol{\ell}}_{s,f(a)})$ consists of a d-dimensional feature map extracted from the image region in the frame $f(a)$ from the training set, with the center set to the object location of tracking result $\hat{\boldsymbol{\ell}}_{f(a)}$ and the scale to b^r relative to the padded object scale of $\hat{\boldsymbol{\ell}}_{f(a)}$ including a context region. Here $r \in \{\lfloor\frac{1-S}{2}\rfloor, \ldots, \lfloor\frac{S-1}{2}\rfloor\}$, and b is the scale increment factor. All samples are obtained by feature computation after the corresponding padded image regions are resized according to b^r and the ratio between the scales of $\hat{\boldsymbol{\ell}}_{f(a)}$ and $\hat{\boldsymbol{\ell}}_1$, leading to the same spatial size $H \times W$. At each spatial location $(u, v, s) \in \Omega$, where $\Omega := \{1, \ldots, H\} \times \{1, \ldots, W\} \times \{1, \ldots, S\}$, we have a d-dimensional feature vector and we denote feature layer $l \in \{1, \ldots, d\}$ of $\mathbf{\Phi}(\hat{\boldsymbol{\ell}}_{s,f(a)})$ by $\mathbf{\Phi}^l(\hat{\boldsymbol{\ell}}_{s,f(a)})$. We use a 3-D scalar-valued Gaussian function $y = \{y_s\}_{s=1}^{S}$ defined over the joint scale-position space Ω as the desired correlation output, which includes a label $y_s(u, v)$ for each location in $\mathbf{\Phi}(\hat{\boldsymbol{\ell}}_{s,f(a)})$. To simplify the notation we denote $\mathbf{\Phi}(\hat{\boldsymbol{\ell}}_{s,f(a)})$ and $\mathbf{\Phi}^l(\hat{\boldsymbol{\ell}}_{s,f(a)})$ as $\mathbf{\Phi}_{s,a}$ and $\mathbf{\Phi}_{s,a}^l$ respectively below.

The desired filter h consists of one $H \times W$ correlation filter h^l per feature layer. The correlation response of h on one $H \times W$ sample $\mathbf{\Phi}_{s,a}$ (the desired output is y_s) is given by

$$R_h(\mathbf{\Phi}_{s,a}) = \sum_{l=1}^{d} h^l \star \mathbf{\Phi}_{s,a}^l \ , \tag{30}$$

where $\star$ denotes circular cross-correlation [60]. Similar to [60], we use $\delta_{\tau,\varsigma}$ to denote the translated Dirac delta function $\delta_{\tau,\varsigma}(u, v) = \delta(u-\tau, v-\varsigma)$, and $*$ to denote circular convolution. Then, the desired filter h^l should satisfy that its inner product with each cyclic shift of the feature layer $\mathcal{C}_{u,v}(\mathbf{\Phi}_{s,a}^l) = \mathbf{\Phi}_{s,a}^l *$

$\delta_{-u,-v}$ is as close as possible to the label $y_s(u,v)$, which is equivalent to minimizing

$$\sum_{(u,v)}\left(\sum_{l=1}^{d}\left\langle \mathcal{C}_{u,v}(\mathbf{\Phi}_{s,a}^{l}), h^{l}\right\rangle - y_s(u,v)\right)^{2} = \left\|\sum_{l=1}^{d} h^{l}\star\mathbf{\Phi}_{s,a}^{l} - y_s\right\|^{2} \tag{31}$$

Here convolution with the translated δ function is equivalent to translation $\left(\mathbf{\Phi}_{s,a}^{l} * \delta_{\tau,\varsigma}\right)(u,v) = \mathbf{\Phi}_{s,a}^{l}(u-\tau \;\mathbf{mod}\; H, v-\varsigma \;\mathbf{mod}\; W)$.

Finally, the filter h is obtained by minimizing (31) over all the training examples as follows:

$$\varepsilon_h = \sum_{a=1}^{|A|}\left(\alpha_a\left(\sum_{s=1}^{S}\left\|R_h(\mathbf{\Phi}_{s,a}) - y_s\right\|^{2}\right) + \lambda\tilde{\alpha}_a\sum_{l=1}^{d}\left\|w\cdot h^{l}\right\|^{2}\right) \tag{32}$$

where the weight $\alpha_a \geq 0$ determines the impact of each training sample from frame $f(a)$, $\lambda \geq 0$ is the weight of the regularization term, w is the Tikhonov regularization weights defined as in [18], and $\tilde{\alpha}_a$ can be set to $\frac{1}{|A|}$ (see Section 4.2 for more details). Note that the weights in w determine the importance of the filter coefficients in h^l, and the coefficients residing inside the padded background context region are suppressed by assigning higher weights in w. The above linear least squares problem can be solved efficiently in the Fourier domain using Parseval's theorem. The discrete Fourier transformed (DFT) filters $\bar{h}^l = \mathcal{F}\{h^l\}$ can then be obtained, where the bar denote the DFT of a function.

3.3.2 Generating unlabeled samples and final fusion.

Once the desired correlation filter h is trained based on the tracking results $\{\hat{\boldsymbol{\ell}}_{f(a)}\}_{a=1}^{|A|}$ in the training set A and updated up to the $(t-1)$-st frame, we are ready to generate a response map in the current t-th frame. Similar to most of the CFs-based trackers, we extract the test samples $\{\mathbf{\Psi}(\hat{\boldsymbol{\ell}}_{s,t-1})\}_{s=1}^{S}$ from the corresponding image regions in frame t, each of which has the center set to the previous location of $\hat{\boldsymbol{\ell}}_{t-1}$ and the scale to b^r relative to the padded object scale of $\hat{\boldsymbol{\ell}}_{t-1}$. We denote the test sample $\mathbf{\Psi}(\hat{\boldsymbol{\ell}}_{s,t-1})$ at resolution s as $\mathbf{\Psi}_s$. Then, the response map $\{R_h(\mathbf{\Psi}_s)\}_{s=1}^{S}$ can be generated efficiently by some operations such as DFT, inverse DFT and point-wise multiplication. This map provides an initial hypothesis for the tracking result in frame t. The result can be determined by obtaining the highest maximal response score with the sub-grid interpolation strategy applied as in the SRDCF tracker [18]. However, this initial hypothesis does not fully exploit the relationships between the historic tracking results and the current frame. In our formulation, we propose to integrate SRDCF with our GPs-based observation likelihood model. The interactions between these two ingredients are two-fold.

First, given the importance of the approximately correct distribution of unlabeled samples, we generate the particles corresponding to these unlabeled samples based on the response maps of CFs. The deployment of this procedure is realized by using the direct simulation method and detailed in step 2 of Algorithm 1. Second, we note that the weight α_a in Eq. (32) is critical for the effectiveness of trained correlation filter h. Surprisingly, the transfer learning extension of GPR in our proposed observation model has provided us with some re-weighted knowledge $\hat{\mathbf{z}}_A$ in Eq. (17) for the auxiliary samples, each of which corresponds with a latent variable $\hat{z}_j$ indicating how this sample is related to the current tracking task. For each auxiliary frame, we collect only one positive auxiliary sample based on the tracking result, and hence the variable $\hat{z}_j$ corresponding with this sample can be used to define the weight α_a of this auxiliary frame in training CFs (see Fig. 2). We only use the auxiliary frames for training CFs, and the weights are updated using the latent variables of positive auxiliary samples, normalized to ensure $\sum_{a=1}^{|A|}\alpha_a = 1$.

Algorithm 1: Tracking-by-Fusion via GPR Extended to Transfer Learning

```
Input: Target sample set $\mathcal{D}_T$, auxiliary sample set $\mathcal{D}_A$, frame
       $t$, latest updated correlation filter $h_{t-1}$, previous
       tracking result $\hat{\boldsymbol{\ell}}_{t-1}$ of frame $t-1$.
Output: updated correlation filter $h_t$, tracking result $\hat{\boldsymbol{\ell}}_t$.
1 Obtain the response map $\{R_{h_{t-1}}(\mathbf{\Psi}_s)\}_{s=1}^{S}$ in the frame $t$
   based on $h_{t-1}$ and $\hat{\boldsymbol{\ell}}_{t-1}$ to approximate the correct
   distribution of unlabeled samples
2 Generate particles $\{\boldsymbol{\ell}_t^i\}_{i=1}^{n_U}$ and the corresponding unlabeled
   samples $\mathcal{X}_U$ based on $\{R_{h_{t-1}}(\mathbf{\Psi}_s)\}_{s=1}^{S}$:
  begin
      At iteration $iter = 0$, initialize each particle $\boldsymbol{\ell}_{t,0}^i \sim$
        $p(\boldsymbol{\ell}_{t,0}|\hat{\boldsymbol{\ell}}_{t-1}) = \mathcal{N}(\boldsymbol{\ell}_{t,0}; \hat{\boldsymbol{\ell}}_{t-1}, \mathbf{\Theta})$ where $i = 1, 2, \ldots, n_U$
      repeat
          $iter = iter + 1$, and set the particle count $num = 0$
          repeat
              Sample a particle $i \sim \mathrm{Uniform}(1, 2, \ldots, n_U)$
              Sample a proposal
                $\boldsymbol{\ell}_t^* \sim p(\boldsymbol{\ell}_{t,iter}|\boldsymbol{\ell}_{t,iter-1}^i) = \mathcal{N}(\boldsymbol{\ell}_{t,iter}; \boldsymbol{\ell}_{t,iter-1}^i, \mathbf{\Theta})$
              Assess the likelihood $R_{h_{t-1}}^*$ given $\boldsymbol{\ell}_t^*$ according
                to $\{R_{h_{t-1}}(\mathbf{\Psi}_s)\}_{s=1}^{S}$ and the translation and
                scale relationships between $\boldsymbol{\ell}_t^*$ and $\hat{\boldsymbol{\ell}}_{t-1}$
              Sample $\theta \sim \mathrm{Uniform}(0, 1)$
              If $\theta < R_{h_{t-1}}^*$, accept the proposal, set $num =$
                $num + 1$, and set $\boldsymbol{\ell}_{t,iter}^{num} = \boldsymbol{\ell}_t^*$
          until $num = n_U$
      until $iter = \mathtt{Threshold}$
3 Construct and solve the GPs-based target task solution of
   Eq. (16), and obtain $\hat{\mathbf{z}}_A$ and the target decision $\hat{\mathbf{z}}_U$
4 Update the correlation filter to $h_t$ by defining the weight $\alpha_a$
   based on $\hat{\mathbf{z}}_A$ and the response map to $\{R_{h_t}(\mathbf{\Psi}_s)\}_{s=1}^{S}$
5 Re-evaluate the unlabeled samples and make the auxiliary
   decision by assessing the likelihood $\mathbf{R}_{h_t} = [R_{h_t}^1, \ldots,$
   $R_{h_t}^{n_U}]^\top$ given $\{\boldsymbol{\ell}_t^i\}_{i=1}^{n_U}$ according to $\{R_{h_t}(\mathbf{\Psi}_s)\}_{s=1}^{S}$ and the
   translation and scale relationships between $\boldsymbol{\ell}_t^i$ and $\hat{\boldsymbol{\ell}}_{t-1}$
6 Fusing two decisions, "pool" is the size of candidate pool
  begin
      $[\cdot, \mathrm{Idx}_A] = \mathtt{sort}(\mathbf{R}_{h_t}, \mathtt{'descend'})$
      $[\cdot, \mathrm{Idx}_T] = \mathtt{sort}(\hat{\mathbf{z}}_U, \mathtt{'descend'})$
      $V_A = \mathrm{Idx}_A(1:\mathtt{pool}) \setminus \{i : \mathrm{Idx}_A(i) \notin \mathrm{Idx}_T(1:\mathtt{pool})\}$
      $V_T = \mathrm{Idx}_T(1:\mathtt{pool}) \setminus \{i : \mathrm{Idx}_T(i) \notin \mathrm{Idx}_A(1:\mathtt{pool})\}$
      if $|V_A| > \mathtt{pool}/2$ then $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t^{V_A(1)}$
      else if $|V_A| = 0$ then $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t^{\mathrm{Idx}_A(1)}$
      else $\hat{\boldsymbol{\ell}}_t = \boldsymbol{\ell}_t^{V_T(1)}$
```

Finally, when the CFs are updated in the current frame, we update the response map and re-evaluate the unlabeled samples by finding the correspondence between the response values and the locations and scales of the corresponding particles. This new tracking task solution, namely auxiliary task solution, is fused with the GPs-based target task solution to make the final decision based on the MAP estimate Eq. (4) of SIR. A heuristic fusion strategy is adopted. Specifically, when obtaining two positive candidate sets according to these two solutions separately, we check the two sets' coincidence degree, e.g., $|V_A|$ in Algorithm 1. When the degree is high, it does not matter whether we rely on the auxiliary decision or the target decision; when the degree is small, we rely more on the target decision to ensure the consistency of the tracking results; when the degree is zero, we rely more on the auxiliary decision to recover from the severe appearance variation and

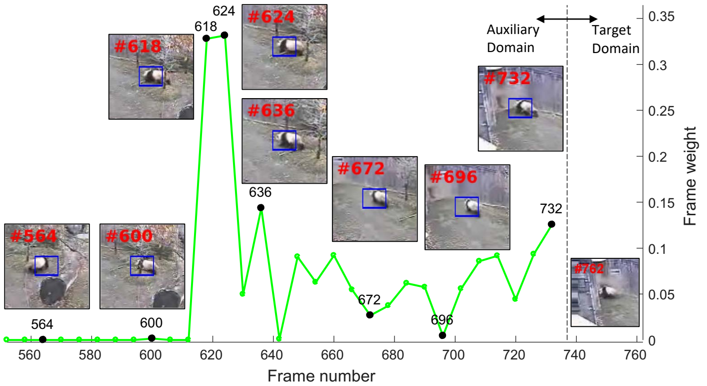

Fig. 2. An illustration of the un-normalized auxiliary frame weights for training CFs at the exemplary frame instance $\#762$ on the example tracking sequence *Panda* in the OTB-2015 benchmark. The weights are obtained from the re-weighting of our GPs-based transfer learning formulation. The image patches in the auxiliary domain are shown for example frames, which are obtained from the auxiliary frames by padding the corresponding tracking results' scales (blue box) to include the context regions. The aim of our tracking is to estimate the location and scale of the panda at frame $\#762$ in the target domain. It can be easily found that the auxiliary frames mostly related to the current tracking task (frames $\#618$ and $\#624$) are considerably up-weighted for training CFs, while the others are down-weighted or even useless (frames $\#564$ and $\#600$).

heavy occlusion. We outline this procedure in Algorithm 1.

4 Experiments

The principal aim of our experiments is to investigate the effectiveness of incorporating CFs into our GPs-based observation likelihood model, where the interactions between the integrated CFs and GPs lead to a new transfer learning based formulation for tracking-by-fusion. To validate that this formulation yields results comparable with recent state-of-the-art trackers, we conduct extensive experiments on four benchmarks, i.e., OTB-2015 [12], Temple-Color [13], and VOT2015/2016 [14], [15], by integrating SRDCF into our formulation. The results are also compared with some baselines and variants of our approach.

Section 4.1 describes the used benchmark datasets with corresponding evaluation criteria and the platform for experimental evaluation. In Section 4.2, we present the details about fusion settings, features, samples collection, and parameters in our experiments. Note that all our settings are fixed for all experiments. The setup of the baselines and our variants as well as the comparisons for illustrating the properties of our tracking formulation are given in Section 4.3. Finally, Section 4.4 presents the comparison with state-of-the-art trackers. More detailed experimental results are also given as the supplementary material.

4.1 Experimental Setup

We evaluate the proposed tracking formulation thoroughly over OTB-2015, Temple-Color, and VOT2015/2016 by following rigorously the evaluation protocols.

The OTB-2015 dataset can be generalized to two sub-benchmarks, namely TB-100 and TB-50. TB-100 includes all the 100 objects in the 98 challenging image sequences, while 50 difficult and representative ones are selected to constitute TB-50 for an in-depth analysis since some of the objects in TB-100 are similar or less challenging. The tracking results on the OTB dataset are reported in terms of the mean overlap precision (OP) and the mean distance precision (DP). The OP score is the fraction of frames in a sequence where the intersection-over-union overlap of the predicted and ground truth rectangles exceeds a given threshold; the DP score is the fraction of frames where the Euclidean distance between the predicted and ground truth centroids is smaller than a given threshold. Based on these two evaluation metrics, the OTB-2015 benchmark provides two kinds of plots to quantify the performance of the trackers. i) In the success plot, the success rate refers to the mean OP over all sequences in each sub-benchmark and is plotted against a uniform range of some thresholds between 0 and 1. An area-under-the-curve (AUC) criterion can also be computed from this success plot. ii) In the precision plot, the precision refers to the mean DP over all sequences in each sub-benchmark and is plotted against a uniform range of some thresholds between 0 and 50 (pixels). Except for One-Pass Evaluation (OPE) which evaluates trackers by running them until the end of a sequence (no-reset) with initialization from the ground truth in the first frame, there are two other evaluation criteria to analyse a tracker's robustness to initialization: Temporal Robustness Evaluation (TRE) and Spatial Robustness Evaluation (SRE). TRE starts the tracker at 20 different frame snapshots while SRE initializes the tracker with perturbed bounding boxes. The Temple-Color benchmark compiles a large set of 128 color sequences to demonstrate the benefit of encoding color information for tracking. It uses the same evaluation protocol to OTB-2015.

In contrast, the VOT2015/2016 benchmarks have many different but more challenging sequences than the aforementioned benchmarks and a reset-based methodology is applied in the toolkit. Whenever a failure (zero overlap of the predicted and ground truth rectangles) is detected, the tracker is re-initialized five frames after the failure. Thus, two weakly correlated performance measures can be used: the accuracy (A) measures how well the predicted bounding box overlaps with the ground truth, i.e., the average overlap computed over the successfully tracked frames; the robustness (R) is estimated by considering how many times the trackers failed during tracking, i.e., the failure rate measure (R_{fr}) computed as the average number of failures, or the reliability (R_S) which can be interpreted as a probability that the tracker will still successfully track the object up to S frames since the last failure

$$R_S = \exp\left(-S\frac{R_{fr}}{N_{frames}}\right),$$

where N_{frames} is the average length of the sequences. Then, the A-R_S pair can be visualized as a 2-D scatter AR-raw plot. The trackers can be ranked with respect to each measure in this plot separately, leading to another AR-rank plot. To quantitatively reflect both robustness and accuracy in a more principled manner while ranking the trackers, the expected average overlap (EAO) measure is proposed to measure the expected no-reset average overlap (AO) of a tracker run on a short-term sequence, although it is computed from the VOT reset-based methodology. In addition, the VOT2016 benchmark supports the OTB no-reset OPE evaluation to measure the true no-reset AO of a tracker. The VOT tracking speed (frames per second) is reported in equivalent filter operation (EFO) units (frames per unit). It is computed by dividing the measured tracking time for a whole sequence with the time required for a predefined filtering operation and then dividing the frame number of the sequence with the computed new tracking time in EFO units.

All experiments are performed on a workstation with Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz, and the running time is about 3 fps. The Matlab code and raw results will be available at `https://github.com/Amgao/TGPRfSRDCF`.

TABLE 1
Ablation Study of the Proposed Tracking-by-Fusion Formulation; We Conduct the Experiments in Terms of OPE on the OTB-2015 Benchmark; the Results Are Reported as Mean OP (%) / Mean DP (%) Scores at Thresholds of 0.5 / 20 Pixels Respectively; We Select SRDCF and Our Initial Work TGPR with HOG Settings as Baselines Since Our Formulation Is Implemented by Fusing SRDCF with TGPR; Two Kinds of Simplified Variants Are Compared: i) TGPRfSRDCF_D without (w/o) Distribution Adaptation and ii) TGPRfSRDCF_W w/o Using the Re-weighted Knowledge; Five Different Learning Rate Values for TGPRfSRDCF_W Are Tested, i.e., 0.015, 0.02, 0.025, 0.03, 0.035.

	TGPR HOG [61]	SRDCF [18]	TGPRfSRDCF_D	TGPRfSRDCF_W					TGPRfSRDCF
Learning Rate	-	0.025	-	0.015	0.02	0.025	0.03	0.035	-
TB-100	63.9/69.7	72.8/78.9	73.1/77.2	78.6/81.4	78.0/81.0	76.8/80.4	77.9/82.0	75.9/79.1	78.5/81.9
TB-50	53.5/59.7	66.6/73.2	65.5/70.7	70.9/74.9	71.4/75.6	70.2/74.9	70.7/76.9	68.3/72.6	73.5/78.4

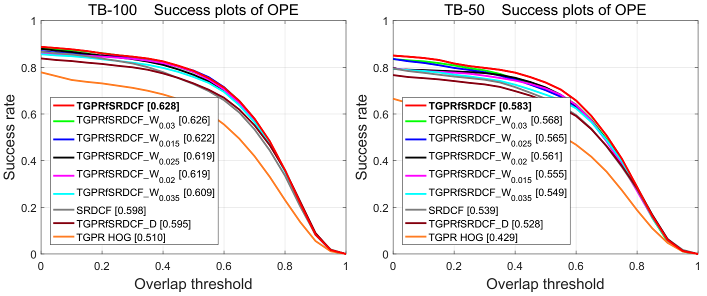

Fig. 3. Success plots showing an ablation study comparison of our proposed formulation with two kinds of variants and some baselines in terms of OPE on the OTB-2015 benchmark. The legend contains the AUC scores for each method. Best viewed in color.

4.2 Implementation Details

Fusion with SRDCF. We explore the representative CFs-based tracker SRDCF for building our blocks, leading to a new tracker named TGPRfSRDCF. Specifically, if we set $\tilde{\alpha}_a = \frac{1}{|A|}$ and $S = 1$ in Eq. (32), then it degenerates to SRDCF, where a 2-D translation filter is learnt and applied to different resolutions to generate the response map over the translation and multi-resolution scale spaces. We borrow the publicly available code of SRDCF for integration, and follow the original settings.

Features and samples collection. We use HOG of the version in [67] for image representation of both GPs and CFs. For generating particles $\{\boldsymbol{\ell}_t^i\}_{i=1}^{n_U}$ from the current frame t, we only consider the variations of 2-D translation ($\vec{x}_t, \vec{y}_t$) and scale (s_t) in the affine transformation. We set the number n_U of particles to 300, and the covariance of $\{\vec{x}_t, \vec{y}_t, s_t\}$ in $\boldsymbol{\Theta}$ to $\{\sigma_x, \sigma_y, 0.05\}$, where $\sigma_x = \sigma_y = \mathbf{max}(8, \mathbf{min}(10, (\mathtt{width}_{t-1} + \mathtt{height}_{t-1})/8))$, $\mathtt{width}_{t-1}$ and $\mathtt{height}_{t-1}$ are the width and height of previous tracking result $\hat{\boldsymbol{\ell}}_{t-1}$. As for $\mathcal{D}_T$, we use the tracking results of past 24 frames $t-24, \ldots, t-1$ (or less than 24 at the beginning of the track) for extracting positive target samples. The negative target samples are collected from the frame $t-1$ around its tracking result $\hat{\boldsymbol{\ell}}_{t-1}$ using dense sampling method in the sliding region, where the Euclidean distances between the center locations of the sampled negative target samples and $\hat{\boldsymbol{\ell}}_{t-1}$ lie in a certain range $(r/4, r/3)$, where $r = (\mathtt{width}_{t-1}^2 + \mathtt{height}_{t-1}^2)^{\frac{1}{2}}/4$. Then, we randomly sample 100 negative target samples. To update the auxiliary sample set slowly, we collect the auxiliary samples $\mathcal{D}_A$ from the frames before $t-24$ at intervals of 6 frames, if these frames are available. The collection in such frames is the same as the collection of labeled samples in [4], except that we add 4 more negative auxiliary samples along the directions of $\vec{x}$ and $\vec{y}$ symmetrically with respect to the tracking result center of the corresponding frame. We set the size limit of the positive auxiliary sample buffer to 35, and thus the negative auxiliary sample buffer to 280. All those samples used in GPs are obtained by re-sizing the corresponding image regions to templates of size 32×32 and extracting the HOG descriptors with a cell size of 4 pixels, leading to a d-dimensional feature vector for each cell, where $d = 31$. In Eq. (25) of Section 3.2.4, we set $N_{\mathrm{r}} = N_{\mathrm{c}} = 2$ to calculate the weights of $\mathbf{W}$. Thus, each block consists of 16 cells, and the dimension of $\mathbf{h}_i^{pq}$ is 496.

Other parameter settings. In Eq. (25), σ_i^{pq} is calculated from the $7th$ nearest neighbor. The parameter k for controlling the sparsity of $\mathbf{W}$ is set to 50. $\mathbf{G}_{\mathrm{all}}$ is defined by setting $\lambda = 1000$. In Section 3.2.3, γ in Eq. (7) is set to 10, η in Eq. (23) is 0.2, and the number of iterations for calculating $\hat{\mathbf{z}}_A$ from Eq. (23) is 40. In Algorithm 1, $\mathtt{Threshold}$ is set to 10, and $\mathtt{pool}$ is 30.

4.3 Experiment 1: Ablation Study

To provide more insights into the effectiveness of the interactions between the integrated CFs and GPs in our formulation, we explore variants of our approach without the interactions for ablation study. Specifically, these simplified variants include, i) variant abbreviated as TGPRfSRDCF_D, which generates unlabeled samples only using the initialized particles $\{\boldsymbol{\ell}_{t,0}^i\}_{i=1}^{n_U}$ in step 2 of Algorithm 1 without distribution adaptation (adapting them to fit the response maps from CFs); and ii) variant abbreviated as TGPRfSRDCF_W, which updates CFs by defining α_a in step 4 of Algorithm 1 according to the commonly used exponentially decaying weights with a fixed learning rate. We test 5 different learning rate values for TGPRfSRDCF_W, i.e., 0.015, 0.02, 0.025, 0.03, 0.035. In addition, we collect some baselines for comparison, including SRDCF and our previous work TGPR. Note that we substitute the feature representation in the original TGPR with HOGs to align with our new approaches for fair comparison.

We start by summarizing the ablation study in terms of OPE on the OTB-2015 dataset in Table 1 and Fig. 3. Table 1 shows the results in mean DP and mean OP, and Fig. 3 shows the success plots of the participants indexed using the AUC score. We remark that our complete version TGPRfSRDCF further consistently improves the performance over all the two sub-benchmarks by providing a significant mean OP score gain of $5.7 \sim 6.9\%$ and AUC score gain of $3.0 \sim 4.4\%$ compared to the baseline SRDCF tracker when fusing it with our initial work TGPR. It also worth mentioning that all the improvements achieve the highest gains on the most challenging sub-benchmark TB-50.

Table 1 and Fig. 3 also show a comparison of the proposed formulation and its simplified variants. From the comparison we see that, our complete version always performs better than the two kinds of variants, or at least not worse, which shows the benefit of using distribution adaptation for generating unlabeled samples and re-weighted knowledge for the update of CFs. Moreover, the variant without distribution adaptation tends to perform worse than the variants without using the re-weighted knowledge, which suggests that the distribution adaptation is a little more crucial factor than using the re-weighted knowledge, although they both play important roles in our formulation. This may be due to our newly designed formulation which starts with generating approximately correct distribution of unlabeled samples. These generated

TABLE 2
Comparison of Our Proposed Formulation (Complete Version) with Some Participating Algorithms in the OTB-2015 Benchmark and Other Latest State-of-the-Art Trackers; We Conduct the Experiments in Terms of OPE, TRE and SRE on OTB-2015, and Only OPE on Temple-Color; the Results Are Reported as AUC (%) / Mean OP (%, at 0.5) / Mean DP (%, at 20 Pixels) Scores.

	TB-100			TB-50			Temple-Color
	OPE	TRE	SRE	OPE	TRE	SRE	OPE
C-COT [20]	67.1/82.0/89.8	67.4/81.9/88.8	63.0/79.8/86.7	61.4/74.9/84.3	62.9/75.9/86.4	57.9/72.4/82.0	57.4/70.2/78.1
TGPRfSRDCF	62.8/78.5/81.9	63.1/78.3/81.3	57.5/74.9/77.4	58.3/73.5/78.4	58.5/73.5/78.0	51.3/66.8/71.0	54.9/67.5/73.5
DeepSRDCF [53]	63.5/77.9/85.1	64.6/77.9/85.5	59.4/75.4/82.1	56.0/67.6/77.2	59.2/71.0/81.6	52.7/66.7/75.3	53.7/65.2/73.8
SRDCFdecon [19]	62.7/76.6/82.5	62.8/76.3/80.4	57.1/73.1/76.9	56.0/69.7/76.4	56.5/69.5/74.9	50.9/65.4/70.3	53.5/65.6/72.7
SRDCF [18]	59.8/72.8/78.9	61.5/74.9/79.1	56.1/71.1/75.6	53.9/66.6/73.2	56.3/69.3/74.8	50.1/64.1/69.1	51.0/62.0/69.4
Staple [50]	58.1/70.9/78.4	59.7/72.8/77.9	54.5/68.2/74.6	50.9/61.2/68.1	53.0/64.4/70.4	49.4/61.2/68.3	-/-/-
SAMF [69]	55.3/67.4/75.1	58.6/72.0/77.5	52.2/65.1/72.4	46.9/57.1/65.0	52.1/64.1/71.3	45.8/56.4/64.7	-/-/-
MEEM [16]	53.0/62.2/78.1	56.5/68.3/79.5	50.2/59.8/73.0	47.3/54.2/71.2	50.9/60.6/74.7	44.5/51.7/65.4	45.9/56.0/63.9
TGPR HOG [61]	51.0/63.9/69.7	53.8/66.2/71.9	46.9/59.5/64.8	42.9/53.5/59.7	47.6/58.9/64.7	39.2/49.7/55.4	41.8/52.2/58.4
DSST [57]	51.3/60.1/68.0	53.8/64.4/69.6	48.5/59.8/65.8	45.2/53.8/60.4	47.3/56.6/62.8	42.2/51.8/58.5	40.7/47.3/53.5
DLR [46]	49.6/58.5/67.3	-/-/-	-/-/-	43.4/49.9/59.7	-/-/-	-/-/-	-/-/-
KCF [17]	44.6/51.0/65.5	51.3/61.1/71.1	43.4/50.9/63.9	35.3/38.3/54.3	43.3/50.6/62.6	35.5/39.6/53.9	38.4/46.1/54.9
Struck [9]	46.2/52.0/63.9	52.1/61.2/70.2	44.0/50.7/61.8	38.2/41.1/53.7	44.9/51.2/61.7	37.3/41.8/53.7	44.1/51.4/61.2
TLD [38]	42.7/50.2/59.6	44.6/51.9/60.6	40.4/47.4/55.9	36.2/42.0/49.5	38.1/43.2/52.0	34.2/40.0/47.6	-/-/-
MIL [6]	33.3/33.4/44.2	39.7/43.5/52.3	32.7/33.1/44.1	26.5/24.4/35.1	31.9/33.1/42.6	26.2/25.3/35.9	33.4/35.7/44.6
IVT [26]	31.6/36.6/43.1	36.9/42.4/47.7	29.2/34.5/40.3	23.7/27.8/32.6	27.7/31.2/36.6	21.5/25.6/30.2	28.9/33.1/41.1

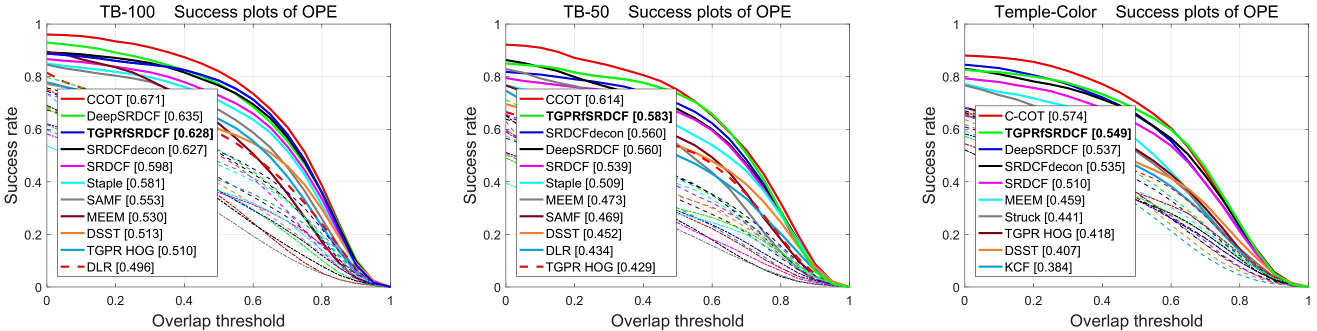

Fig. 4. Success plots showing the performance of our formulation (complete version) compared to some representative tracking algorithms provided with OTB-2015 and some latest state-of-the-art trackers in terms of OPE on the OTB-2015 and Temple-Color benchmarks. The legends of the success plots contain the AUC scores for each method. Only some top-performing trackers are displayed in the legend for clarity. Best viewed in color.

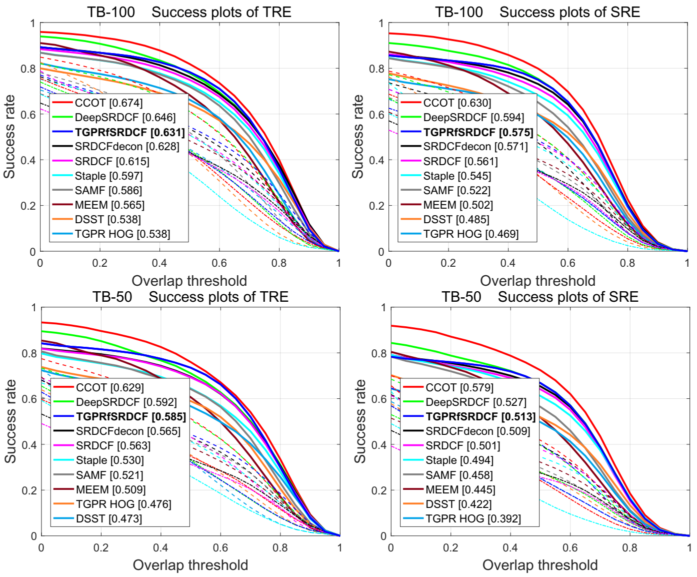

Fig. 5. Success plots showing the performance of our formulation (complete version) compared to some representative tracking algorithms provided with OTB-2015 and some latest state-of-the-art trackers in terms of TRE and SRE on the OTB-2015 benchmark. The legends of the success plots contain the AUC scores for each method. Only some top-performing trackers are displayed in the legend for clarity. Best viewed in color.

unlabeled samples influence not only the auxiliary/target task solution, but also the learning of the re-weighted knowledge. The comparison with the variants using the exponentially decaying weights with 5 different learning rates also shows that our automatically learnt re-weighted knowledge can avoid using the tediously hand-tuned different learning rates on different benchmarks to achieve the superior results.

4.4 Experiment 2: State-of-the-Art Comparison

In this section, we conduct a comprehensive comparison of our proposed formulation (complete version) with some representative tracking algorithms provided with the OTB-2015 benchmark including MIL [6], IVT [26], TLD [38], Struck [9], etc., and also with some latest state-of-the-art trackers including MEEM [16], our initial work TGPR [61] with HOGs, MDNet [24], EBT [70], SiamFC [25], DLR [46], and a large family of CFs-based trackers, i.e., KCF [17], DSST [57], SAMF [69], Staple [50], SRDCF [18], SRDCFdecon [19], DeepSRDCF [53], C-COT [20].

C-COT was appraised as the best tracker on VOT-2016 [15] for its significant innovation in relaxing the constant feature map dimension assumption of SRDCF and allowing the spatially regularized CFs to be learnt on the feature maps of multiple different resolutions from the pre-trained CNNs. This breakthrough enhances the effectiveness of using the deep feature maps at different layers and boosts the tracking performance significantly in the literature. DeepSRDCF can be seen as a simplified single-resolution version of C-COT when using the combination of convolutional layers. SRDCFdecon also goes in the direction of improving the baseline SRDCF, and its property of learning continuous weights for decreasing the impact of corrupted samples also exists in our formulation.

OTB-2015 and Temple-Color datasets. We summarize the comparison results by reporting them in terms of AUC, mean OP and mean DP scores in Table 2 and showing the success plots of some top trackers indexed using the AUC score in Figs. 4 and 5. From these results we see that, despite the inferior performance compared to C-COT using the deep features, our complete version TGPRfSRDCF consistently improves

TABLE 3
Comparison of Our Proposed Formulation (Complete Version) with Some Participating Algorithms on the VOT2015/2016 Benchmarks; the Results Are Reported as EAO, A, R_{fr} or R_S ($S = 100$), No-Reset AO, and VOT Tracking Speed in EFO. A Large Value in EFO Indicates a High Tracking Speed.

VOT		SRDCF [18]	MDNet_N [24]	SRDCFdecon [19]	SiamFC_R [25]	DeepSRDCF [53]	Staple [50]	TGPRfSRDCF	EBT [70]	SSAT	TCNN	C-COT [20]
2015	EAO	0.288	-	0.299	-	0.318	0.300	0.311	0.313	-	-	0.303
	A	0.551	-	0.552	-	0.562	0.563	0.572	0.453	-	-	0.535
	R_{fr}	1.242	-	1.095	-	1.046	1.385	1.254	1.021	-	-	0.821
2016	EAO	0.247	0.257	-	0.277	0.276	0.295	0.279	0.291	0.321	0.325	0.331
	A	0.536	0.542	-	0.550	0.529	0.547	0.551	0.465	0.579	0.555	0.541
	$-\ln R_S$	0.419	0.337	-	0.382	0.326	0.378	0.376	0.252	0.291	0.268	0.238
	AO	0.398	0.458	-	0.422	0.428	0.390	0.410	0.370	0.516	0.487	0.470
	EFO	1.990	0.534	-	5.444	0.380	11.114	1.563	3.011	0.475	1.049	0.507

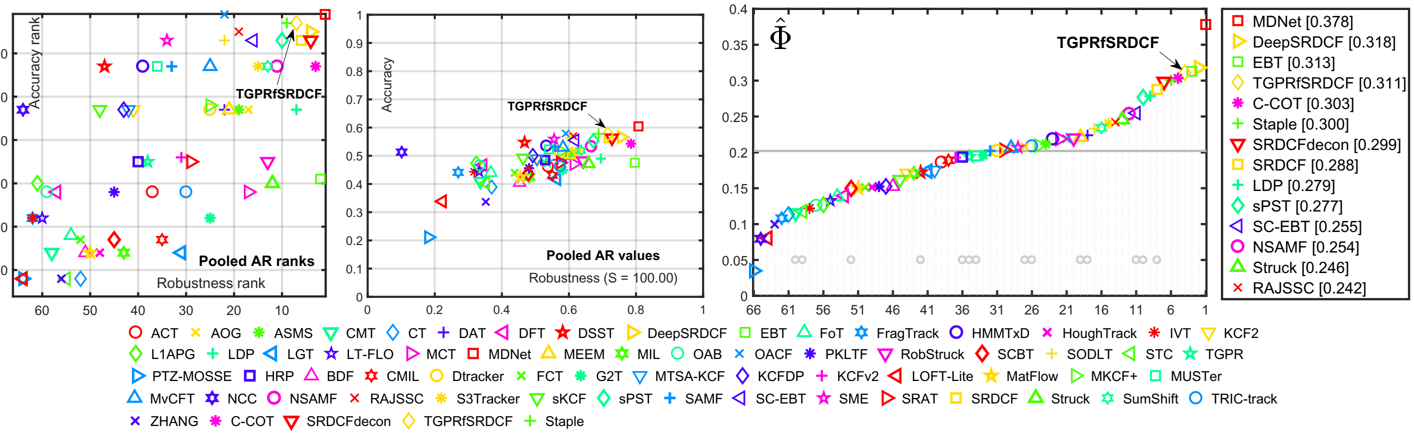

Fig. 6. The AR-rank plot (left) and AR-raw plot (middle) using the A-R_S pairs generated by sequence pooling on VOT2015, and the corresponding expected average overlap graph (right) with trackers ranked from right to left. Best viewed in color.

SRDCF in terms of OPE, TRE and SRE over all the TB-100, TB-50 and Temple-Color benchmarks as other improved versions (i.e., C-COT, DeepSRDCF and SRDCFdecon) have done. We also expect consistent improvement over C-COT when integrating C-COT into our formulation. TGPRfSRDCF achieves comparable performance with DeepSRDCF, and even superior performance in terms of AUC of OPE over the more challenging TB-50 and Temple-Color benchmarks by providing gains ranging from $1.2 \sim 2.3\%$. As for SRDCFdecon, TGPRfSRDCF significantly outperforms it in terms of all the performance criteria over TB-50 and Temple-Color.

In addition, it is worth noting that in Table 2 TGPRfSRDCF mostly outperforms DeepSRDCF in terms of mean OP at the threshold of 0.5 while being inferior with respect to AUC and mean DP. This is consistent with the observation in Figs. 4 and 5, where DeepSRDCF always achieves much higher mean OP scores at the lower overlap thresholds. The reason is that DeepSRDCF can always at least capture part of the object while encountering large object variations so that the overlap with the ground truth is always above zero and the distance to the center of the ground truth below 20 pixels. This can be attributed to the deep convolutional features learnt from the large ImageNet dataset for object detection and classification. These features can capture different colors and edges over image regions (the feature dimension is 96), and hence more robust to object variations including deviation from the predicted location.

VOT2015/2016 datasets. We show the comparative results on VOT2015/2016 in Table 3 and Figs. 6, 7 and 8. Among the compared methods, TGPRfSRDCF achieves favorable results in terms of accuracy (see the AR-rank plots of Figs. 6 and 7), while still being comparable with the top-performing trackers C-COT, TCNN, SSAT, MDNet (or MDNet_N), EBT, Staple and DeepSRDCF in terms of the other measurements. Note that MDNet_N and SSAT on VOT2016 are variations of the VOT2015 winner MDNet which is derived from CNN, except that MDNet_N eliminates the multi-domain pre-training process using other tracking datasets. Further more, TCNN extends MDNet_N to a tree-structured appearance model for tracking.

In addition, TGPRfSRDCF significantly exceeds the baseline tracker SRDCF without sacrificing much tracking speed. TGPRfSRDCF also exceeds the VOT2015 and VOT2016 published state-of-the-art bounds (gray horizontal lines in the expected average overlap graphs of Figs. 6 and 7), which demonstrates the superiority of our proposed formulation.

5 Conclusion

In this work, we demonstrated that it is possible to directly analyse the probability density function for target appearance in the sequential Bayesian inference based tracking framework using GPs. The resulting transfer learning based tracking formulation provides an interface for tracking-by-fusion with CFs. Specifically, the transfer learning extension of GPR receives the response maps of CFs as the approximately correct distribution for generating unlabeled samples, while in return providing more conscious re-weighted knowledge to CFs for updating. We have applied this new formulation to integrate the baseline tracker SRDCF, leading to significant improvement and state-of-the-art performances.

Since the preliminary version of this work, learning continuous weights for decreasing the impact of corrupted samples has been exploited successfully for CFs based tracker in [19]. C-COT was appraised as the best tracker on VOT-2016 for its significant innovation in relaxing the constant feature map dimension assumption. An interesting direction for further work is to integrate C-COT into our formulation to gain further improvement. Another interesting direction for further work is to learn affinity in Eq. (25) using deep learning and directly output all the similarities in a purely data-driven manner [71], instead of designing the similarity kernels on image features.

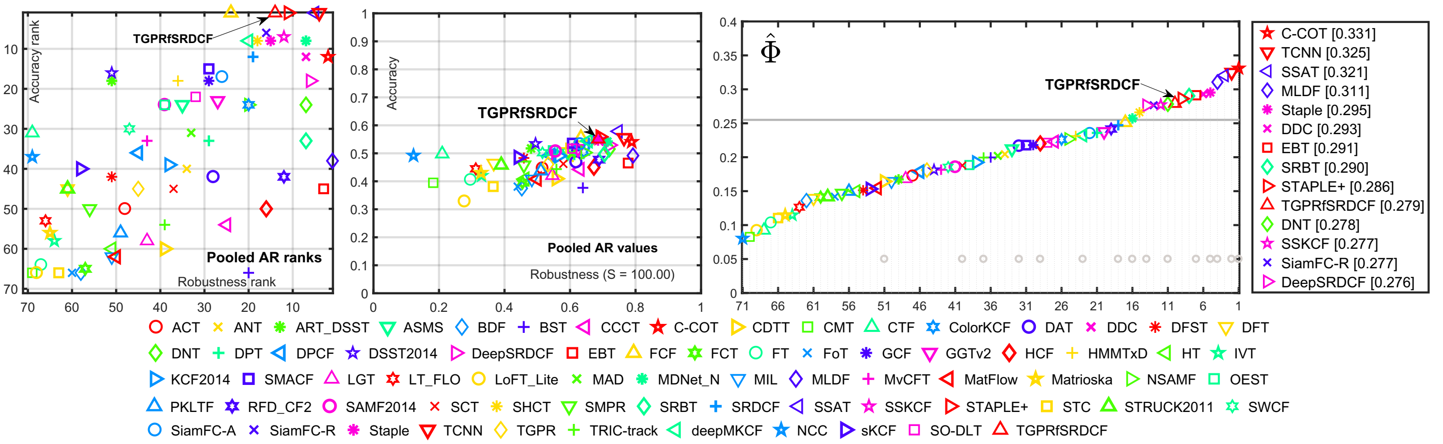

Fig. 7. The AR-rank plot (left) and AR-raw plot (middle) using the A-R_S pairs generated by sequence pooling on VOT2016, and the corresponding expected average overlap graph (right) with trackers ranked from right to left. Best viewed in color.

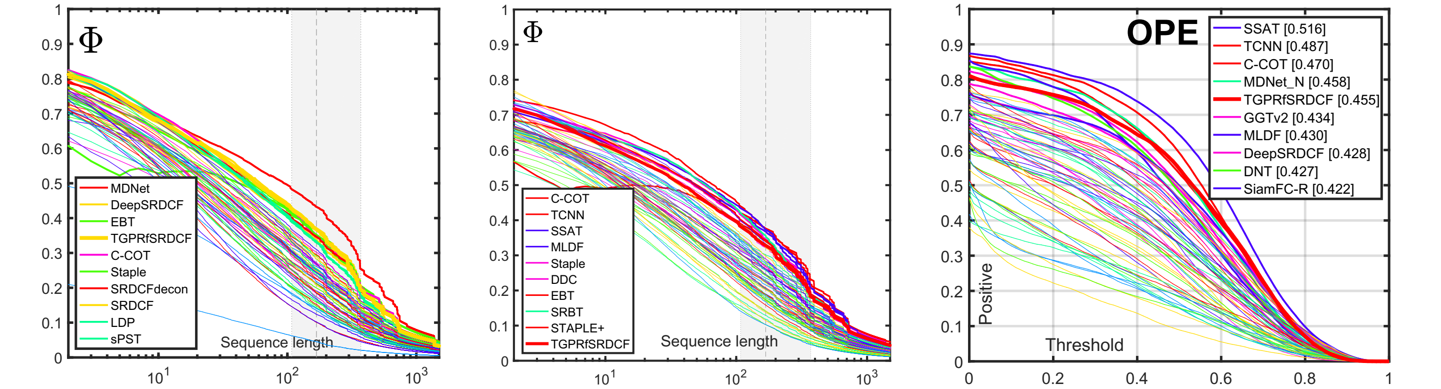

Fig. 8. Expected average overlap curves on VOT2015 (left) and VOT2016 (middle), and the OPE no-reset plot for AO curves on VOT2016 (right). Best viewed in color.

Appendix A

A.1 Partitioned Matrix Inversion Lemma

In a special case of partitioned matrix inversion lemma, let a symmetric, positive semidefinite matrix $\mathbf{S}$ and its inverse $\mathbf{S}^{-1}$ be partitioned into

$$\mathbf{S} = \begin{array}{c} \overbrace{\quad}^{p}\overbrace{\quad}^{q} \\ \left[\begin{array}{cc} \mathbf{C} & \mathbf{O} \\ \mathbf{O}^\top & \mathbf{D} \end{array}\right] \end{array} \begin{array}{l} \\ \}\,p \\ \}\,q \end{array} \qquad \mathbf{S}^{-1} = \begin{array}{c} \overbrace{\quad}^{p}\overbrace{\quad}^{q} \\ \left[\begin{array}{cc} \mathbf{J} & \mathbf{Q} \\ \mathbf{Q}^\top & \mathbf{L} \end{array}\right] \end{array} \begin{array}{l} \\ \}\,p \\ \}\,q \end{array}$$

then we have

$$\mathbf{L} = \left(\mathbf{D} - \mathbf{O}^\top\mathbf{C}^{-1}\mathbf{O}\right)^{-1}, \tag{33}$$

$$\mathbf{Q} = -\mathbf{C}^{-1}\mathbf{O}\mathbf{L}\,, \tag{34}$$

$$\mathbf{J} = \left(\mathbf{C} - \mathbf{O}\mathbf{D}^{-1}\mathbf{O}^\top\right)^{-1} = \mathbf{C}^{-1} + \mathbf{O}\mathbf{L}^{-1}\mathbf{O}^\top\,. \tag{35}$$

A.2 Proof of Proposition 1

Taking the logarithm of Eq. (10), we have

$$\begin{aligned} \ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\right) &= \ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U, \mathbf{y}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)\right) \\ &\quad - \ln\left(p\left(\mathbf{y}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)\right) \end{aligned} \tag{36}$$

Denote $\mathbf{z} = \begin{pmatrix} \mathbf{z}_A \\ \mathbf{z}_U \end{pmatrix}$ and $\mathbf{z}^\top = \begin{pmatrix} \mathbf{z}_A^\top & \mathbf{z}_U^\top \end{pmatrix}$. Recall $\mathbf{G}_{\text{all}}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{M} \end{pmatrix}$ and let $\mathbf{G}_{\text{all}} = \begin{pmatrix} \mathbf{G}_{TT} & \mathbf{G}_{TZ} \\ \mathbf{G}_{ZT} & \mathbf{G}_{ZZ} \end{pmatrix}$, then

$$\begin{aligned} &\ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U, \mathbf{y}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)\right) \\ &= -\frac{1}{2}\left(\ln(2\pi)^{n_T+n_A+n_U} + \ln|\mathbf{G}_{\text{all}}| + \begin{pmatrix} \mathbf{y}_T^\top & \mathbf{z}^\top \end{pmatrix} \mathbf{G}_{\text{all}}^{-1} \begin{pmatrix} \mathbf{y}_T \\ \mathbf{z} \end{pmatrix}\right), \end{aligned}$$

$$\begin{aligned} &\ln\left(p\left(\mathbf{y}_T | \mathcal{X}_A, \mathcal{X}_U, \mathcal{X}_T\right)\right) \\ &= -\frac{1}{2}\left(\ln(2\pi)^{n_T} + \ln|\mathbf{G}_{TT}| + \mathbf{y}_T^\top\mathbf{G}_{TT}^{-1}\mathbf{y}_T\right), \end{aligned}$$

where $\mathbf{G}_{TT} = (\mathbf{A} - \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\top)^{-1}$. Because

$$\begin{aligned} &\ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\right) \\ &= -\frac{1}{2}\left(\ln(2\pi)^{n_A+n_U} + \ln|\mathbf{G}| + (\mathbf{z}-\boldsymbol{\mu})^\top \mathbf{G}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right), \end{aligned}$$

it is then trivial to follow the partitioned matrix inversion lemma in Appendix A.1 to prove that $\boldsymbol{\mu} = -\mathbf{M}^{-1}\mathbf{B}^\top\mathbf{y}_T$ and $\mathbf{G} = \mathbf{M}^{-1}$ satisfy Eq. (36). Thus, the logarithm of Eq. (10) can also be represented as

$$\ln\left(p\left(\mathbf{z}_A, \mathbf{z}_U | \mathcal{X}_A, \mathcal{X}_U, \mathcal{D}_T\right)\right) = -\frac{1}{2}\left(\begin{pmatrix} \mathbf{y}_T^\top & \mathbf{z}^\top \end{pmatrix} \mathbf{G}_{\text{all}}^{-1} \begin{pmatrix} \mathbf{y}_T \\ \mathbf{z} \end{pmatrix}\right) + c\,,$$

where the constant value $c = -\frac{1}{2}(\mathbf{y}_T^\top(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^\top - \mathbf{A})\mathbf{y}_T + \ln|\mathbf{G}| + \ln(2\pi)^{n_A+n_U})$.

A.3 Proof of Proposition 2

Denote $\mathbf{y} = \begin{pmatrix} \mathbf{y}_T \\ \mathbf{z}_A \end{pmatrix}$ and $\mathbf{y}^\top = \begin{pmatrix} \mathbf{y}_T^\top & \mathbf{z}_A^\top \end{pmatrix}$. Recall $\mathbf{G}_{\text{all}} = \begin{pmatrix} \mathbf{G}_{LL} & \mathbf{G}_{LU} \\ \mathbf{G}_{UL} & \mathbf{G}_{UU} \end{pmatrix}$ and let $\mathbf{G}_{\text{all}}^{-1} = \begin{pmatrix} \mathbf{A}_L & \mathbf{B}_L \\ \mathbf{B}_L^\top & \mathbf{M}_L \end{pmatrix}$. Then, replacing the second term $Q_2(\mathbf{z}_A, \mathbf{z}_U) = -\frac{1}{2}\left(\begin{pmatrix} \mathbf{y}^\top & \mathbf{z}_U^\top \end{pmatrix} \mathbf{G}_{\text{all}}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{z}_U \end{pmatrix}\right) + c$ in Eq. (14) results in

$$\mathcal{J} = \ln\left(\prod_{j=n_T+1}^{n_L} \mathrm{P_r}\left(y_i | z_i\right)\right) - \frac{1}{2}\left(\begin{pmatrix} \mathbf{y}^\top & \mathbf{z}_U^\top \end{pmatrix} \mathbf{G}_{\text{all}}^{-1} \begin{pmatrix} \mathbf{y} \\ \mathbf{z}_U \end{pmatrix}\right) + c\,.$$

$$\begin{aligned}
&\text{Let } \mathrm{z}_U = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_T\\ \mathbf{z}_A\end{pmatrix} = \mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y}\text{, then}\\
&\quad\begin{pmatrix}\mathbf{y}^\top & \mathbf{z}_U^\top\end{pmatrix}\mathbf{G}_{\text{all}}^{-1}\begin{pmatrix}\mathbf{y}\\ \mathbf{z}_U\end{pmatrix}\\
&=\mathbf{y}^\top\mathbf{A}_L\mathbf{y} + \mathbf{z}_U^\top\mathbf{B}_L^\top\mathbf{y} + \mathbf{y}^\top\mathbf{B}_L\mathbf{z}_U + \mathbf{z}_U^\top\mathbf{M}_L\mathbf{z}_U\\
&=\mathbf{y}^\top\mathbf{A}_L\mathbf{y} - 2\mathbf{y}^\top\mathbf{G}_{LL}^{-1}\mathbf{G}_{LU}\mathbf{M}_L\mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y} + \mathbf{z}_U^\top\mathbf{M}_L\mathbf{z}_U\\
&=\mathbf{y}^\top\left(\mathbf{G}_{LL}^{-1} + \mathbf{B}_L\mathbf{M}_L^{-1}\mathbf{B}_L^\top\right)\mathbf{y} - \mathbf{y}^\top\mathbf{G}_{LL}^{-1}\mathbf{G}_{LU}\mathbf{M}_L\mathbf{G}_{UL}\mathbf{G}_{LL}^{-1}\mathbf{y}\\
&=\mathbf{y}^\top\mathbf{G}_{LL}^{-1}\mathbf{y} + \mathbf{y}^\top\mathbf{B}_L\mathbf{M}_L^{-1}\mathbf{B}_L^\top\mathbf{y} - \mathbf{y}^\top\mathbf{B}_L\mathbf{M}_L^{-1}\mathbf{B}_L^\top\mathbf{y}\\
&=\mathbf{y}^\top\mathbf{G}_{LL}^{-1}\mathbf{y}\\
&=\begin{pmatrix}\mathbf{y}_T^\top & \mathbf{z}_A^\top\end{pmatrix}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_T\\ \mathbf{z}_A\end{pmatrix}\ .
\end{aligned}\tag{37}$$

So $\mathcal{J}$ can be written as

$$\mathcal{J} = \sum_{j=n_T+1}^{n_L}\ln\left(\Pr\left(y_i|z_i\right)\right) - \frac{1}{2}\begin{pmatrix}\mathbf{y}_T^\top & \mathbf{z}_A^\top\end{pmatrix}\mathbf{G}_{LL}^{-1}\begin{pmatrix}\mathbf{y}_T\\ \mathbf{z}_A\end{pmatrix} + c\ .$$

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work is supported by Natural Science Foundation of China (Grant No. 61602478, 61751212 and 61472421), Beijing Natural Science Foundation (Grant No. L172051), the NSFC-general technology collaborative Fund for basic research (Grant No. U1636218), the Key Research Program of Frontier Sciences, CAS (Grant No. QYZDJ-SSW-JSC040), the CAS External cooperation key project, and the Research Project of ForwardX Robotics, Inc. Prof. Ling is supported in part by US NSF Grants 1350521, 1618398 and 1814745.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[3] H. Grabner, C. Leistner, and H. Bisshof, “Semi-supervised on-line boosting for robust tracking,” in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

[4] G. Li, L. Qin, Q. Huang, J. Pang, and S. Jiang, “Treat samples differently: Object tracking with semi-supervised online CovBoost,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 627–634.

[5] X. Li, A. Dick, H. Wang, C. Shen, and A. van den Hengel, “Graph mode-based contextual kernels for robust svm tracking,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1156–1163.

[6] B. Babenko, M.-H. Yang, and S. Belongie, “Robust object tracking with online multiple instance learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[7] X. Li, C. Shen, A. Dick, and A. van den Hengel, “Learning compact binary codes for visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2419–2426.

[8] W. Zhong, H. Lu, and M.-H. Yang, “Robust object tracking via sparsity-based collaborative model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1838–1845.

[9] S. Hare, A. Saffari, and P. H. Torr, “Struck: Structured output tracking with kernels,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, Oct. 2016.

[10] W. Hu, J. Gao, J. Xing, C. Zhang, and S. Maybank, “Semi-supervised tensor-based graph embedding learning and its application to visual discriminant tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 172–188, Jan. 2017.

[11] Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2411–2418.

[12] ——, “Object tracking benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sept. 2015.

[13] P. Liang, E. Blasch, and H. Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE Trans. on Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.

[14] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojíř, G. Häger, G. Nebehay, and R. Pflugfelder, “The visual object tracking VOT2015 challenge results,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 564–586.

[15] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojíř, G. Häger, A. Lukězič, and G. Fernández, “The visual object tracking VOT2016 challenge results,” in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2016, pp. 777–823.

[16] J. Zhang, S. Ma, and S. Sclaroff, “MEEM: Robust tracking via multiple experts using entropy minimization,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[18] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, “Learning spatially regularized correlation filters for visual tracking,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4310–4318.

[19] ——, “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.

[20] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, “Beyond correlation filters: Learning continuous convolution operators for visual tracking,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 472–488.

[21] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang, “Real-time visual tracking: Promoting the robustness of correlation filter learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 662–678.

[22] M. Mueller, N. Smith, and B. Ghanem, “Context-aware correlation filter tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1387–1395.

[23] S. Hong, T. You, S. Kwak, and B. Han, “Online tracking by learning discriminative saliency map with convolutional neural network,” in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[24] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4293–4302.

[25] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2016, pp. 850–865.

[26] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” *Int. J. Comput. Vis.*, vol. 77, no. 1-3, pp. 125–141, May 2008.

[27] B. Ristic, N. Gordon, and S. Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. USA: Artech House, 2004.

[28] R. Herbrich, *Learning kernel classifiers: Theory and algorithms*. Cambridge, Mass.: MIT Press, 2001.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Mass.: MIT Press, 2006.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cambridge CB3 0FB, U.K.: Springer, 2006.

[31] M. Steyvers, *Computational Statistics with Matlab*, Univ. of California, Irvine, Calif., 2011. [Online]. Available: http://psiexp.ss.uci.edu/research/teachingP205C/205C.pdf

[32] M. Hürzeler and H. Künsch, “Monte Carlo approximation for general state space model,” *J. Comput. Graph. Statist.*, vol. 7, no. 2, pp. 175–193, 1998.

[33] X. Lan, A. J. Ma, and P. C. Yuen, “Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1194–1201.

[34] Z. Zhang and K. H. Wong, “Pyramid-based visual tracking using sparsity represented mean transform,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1226–1233.

[35] J. Kwon and K. M. Lee, “Visual tracking decomposition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1269–1276.

[36] ——, “Minimum uncertainty gap for robust visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2355–2362.

[37] ——, “Interval tracker: Tracking by interval analysis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3494–3501.

[38] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, July 2012.

[39] Q. Bai, Z. Wu, S. Sclaroff, M. Betke, and C. Monnier, “Randomized ensemble tracking,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2040–2047.

[40] D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, “Multihypothesis trajectory analysis for robust visual tracking,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5088–5096.

[41] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3074–3082.

[42] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4303–4311.

[43] D. Chen, Z. Yuan, G. Hua, Y. Wu, and N. Zheng, "Description-discrimination collaborative tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.

[44] Y. Sui, Y. Tang, and L. Zhang, "Discriminative low-rank tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3002–3010.

[45] Y. Sui, G. Wang, L. Zhang, and M.-H. Yang, "Exploiting spatial-temporal locality of tracking via structured dictionary learning," *IEEE Trans. on Image Process.*, vol. 27, no. 3, pp. 1282–1296, Mar. 2018.

[46] Y. Sui, Y. Tang, L. Zhang, and G. Wang, "Visual tracking via subspace learning: A discriminative approach," *Int. J. Comput. Vis.*, vol. 126, no. 5, p. 515536, May 2018.

[47] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel robust online simple tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 723–730.

[48] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 749–758.

[49] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 5388–5396.

[50] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1401–1409.

[51] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Adv. Neural Info, Proc. Syst. 26*, 2013, pp. 809–817.

[52] L. Wang, W. O. X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3119–3127.

[53] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 621–629.

[54] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2544–2550.

[55] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 702–715.

[56] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 127–141.

[57] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, Sept. 2016.

[58] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, 2015, pp. 595–603.

[59] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1090–1097.

[60] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for Correlation Filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2805–2813.

[61] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.

[62] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1501.04587

[63] K. Meshgi, S. Oba, and S. Ishii, "Efficient diverse ensemble for discriminative co-tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4814–4823.

[64] X. Zhu, J. Lafferty, and Z. Ghahramani, "Semi-supervised learning: From Gaussian fields to Gaussian processes," Carnegie Mellon Univ., Pittsburgh, Penn., Tech. Rep. CMU-CS-03-175, 2003.

[65] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[66] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, "Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2420–2440, Dec. 2012.

[67] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sept. 2010.

[68] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Adv. Neural Info, Proc. Syst. 17*, 2005, pp. 1601–1608.

[69] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2014, pp. 254–265.

[70] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 943 – 951.

[71] S. Liu, S. D. Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, "Learning affinity via spatial propagation networks," in *Adv. Neural Info, Proc. Syst. 30*, 2017, pp. 1520–1530.

Jin Gao received the B.S. degree from the Beihang University, Beijing, China in 2010, and the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS) in 2015. Now he is an assistant professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA). His research interests include visual tracking, autonomous vehicles, and service robots.

Qiang Wang received the B.S. degree in automation from University of Science and Technology Beijing, Beijing, China, in 2015. Now, he is working towards the Ph.D. degree from the Institute of Automation, University of Chinese Academy of Sciences (UCAS). His research interests are both theory and applications of single object tracking.

Junliang Xing received the B.S. degree in Computer Science from Xi'an Jiaotong University, Shaanxi, China, in 2007, and the Ph.D. degree in Computer Science and Technology from Tsinghua University, Beijing, China, in 2012. He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China. His research interests mainly focus on computer vision problems like object detection, tracking, segmentation, and recognition.

Haibin Ling received the BS and MS degrees from Peking University, China, in 1997 and 2000, respectively, and the PhD degree from the University of Maryland, College Park, in 2006. From 2006 to 2007, he worked as a postdoctoral scientist at UCLA. In 2008, he joined Temple University where he is now an associate professor. He received the Best Student Paper Award at ACM UIST in 2003, and the NSF CAREER Award in 2014. He serves as Associate Editors for IEEE T-PAMI, Pattern Recognition, and CVIU. He has also served as Area Chairs for CVPR 2014, CVPR 2016 and CVPR 2019.

Weiming Hu received the Ph.D. degree from the Dept. of Computer Science and Engineering, Zhejiang University, Zhejiang, China. Since 1998, he has been with the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, where he is currently a Professor. He has published more than 200 papers on peer reviewed international conferences and journals. His current research interests include visual motion analysis and recognition of harmful Internet multimedia.

Stephen Maybank received a BA in Mathematics from King's college Cambridge in 1976 and a Ph.D. in computer science from Birkbeck college, University of London in 1988. Now he is a professor in the School of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, etc.He is a fellow of the IEEE and fellow of the Royal Statistical Society.