Journal of Real-Time Image Processing (2024) 21:78
https://doi.org/10.1007/s11554-024-01456-2

RESEARCH

Enhancing UAV tracking: a focus on discriminative representations using contrastive instances

Xucheng Wang[1] · Dan Zeng[2] · Yongxin Li[1] · Mingliang Zou[1] · Qijun Zhao[3] · Shuiwang Li[1]

Received: 18 December 2023 / Accepted: 25 March 2024 / Published online: 21 April 2024

Abstract
Addressing the core challenges of achieving both high efficiency and precision in UAV tracking is crucial due to limitations in computing resources, battery capacity, and maximum load capacity on UAVs. Discriminative correlation filter (DCF)-based trackers excel in efficiency on a single CPU but lag in precision. In contrast, many lightweight deep learning (DL)-based trackers based on model compression strike a better balance between efficiency and precision. However, higher compression rates can hinder performance by diminishing discriminative representations. Given these challenges, our paper aims to enhance feature representations' discriminative abilities through an innovative feature-learning approach. We specifically emphasize leveraging contrasting instances to achieve more distinct representations for effective UAV tracking. Our method eliminates the need for manual annotations and facilitates the creation and deployment of lightweight models. As far as our knowledge goes, we are the pioneers in exploring the possibilities of contrastive learning in UAV tracking applications. Through extensive experimentation across four UAV benchmarks, namely, UAVDT, DTB70, UAV123@10fps and VisDrone2018, We have shown that our DRCI (discriminative representation with contrastive instances) tracker outperforms current state-of-the-art UAV tracking methods, underscoring its potential to effectively tackle the persistent challenges in this field.

Keywords UAV tracking · Contrastive instances · Discriminative representation · Contrastive learning

1 Introduction

UAV tracking as a subset of object tracking, draws considerable attention because of its potential in various applications, such as navigation, agriculture, transportation, disaster response, and public safety [1–6]. UAV tracking focuses on assessing and predicting the location and size of arbitrary targets in continuous aerial imagery, which is a critical capability for tasks ranging from automated monitoring of crop health in precision agriculture to effective coordination during disaster management and surveillance in public safety operations. Despite being a subset of object tracking, UAV tracking presents a variety of unique challenges that impede the achievement of high precision and efficiency. UAV tracking faces challenges such as motion blur, extreme viewing angles, severe occlusion, and changing scales due to high-speed UAV movement. These factors, especially when combined, significantly degrade precision. The finite computational resources aboard UAVs require streamlined algorithms for real-time operation, demanding exceptional efficiency in design and execution. Moreover, stringent power constraints and limited battery capacity impose strict limitations on the duration of tracking operations. The operational window becomes dictated by the available power, posing a continual challenge for sustained tracking tasks [3, 4, 7].

Given these unique obstacles and constraints, the need for innovative tracking techniques increases dramatically. New approaches should not only mitigate these difficulties but should also be able to adapt swiftly to the ever-changing scenarios intrinsic to UAV operations. This demands ongoing innovation in the design of UAV tracking methodologies and technologies. The discriminative correlation filter (DCF) approach has notably excelled in efficiency, reigning supreme on a single CPU. However, its prowess in speed is countered by a relative lag in accuracy compared to the

✉ Shuiwang Li
lishuiwang0721@163.com

[1] College of Computer Science and Engineering, Guilin University of Technology, Guilin 541000, China

[2] Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518000, China

[3] College of Computer Science, Sichuan University, Chengdu 30332, China

cutting-edge deep learning (DL)-based trackers [1, 8–13]. The DL-based trackers are acclaimed for their unparalleled accuracy. Yet, their strength in accuracy often comes at the cost of efficiency due to their reliance on intricate and resource-intensive architectures. To narrow this gap and meet the growing demand for both precision and efficiency, a recent trend has emerged—the development of lightweight DL-based trackers [3, 4, 14, 15]. This evolution represents a concerted effort within the field, aiming to combine the precision of DL-based approaches with streamlined architectures that prioritize computational efficiency. The introduction of lightweight DL-based trackers represents a significant turning point in the methodology of UAV tracking. It represents a shift towards reconciling the balance between accuracy and efficiency, striving to harness the advantages of DL-based precision while mitigating the computational burdens inherent in complex architectures. This endeavor is consistent with the primary objective of enhancing tracking performance within the resource-constrained environment of UAV tracking. These approaches primarily leverage model compression methods, like filter pruning, to enhance efficiency while upholding a strong level of precision. Despite the simplicity of filter pruning methods utilized in works like Fisher pruning [4] and rank-based filter pruning [3]. However, the attained results for tracking precision and efficiency fall short of expectations and remain unsatisfactory. The primary drawback of these approaches stems from the utilization of high compression rates, which often result in subpar discriminative representations [16, 17]. In light of this, our paper explores a new feature-learning approach designed to address the challenge of low performance in UAV tracking. The primary objective is to augment the discriminative capabilities of feature representations.

Contrastive learning functions as a discriminative and representation learning technique, aiming to construct an embedding space where similar sample sets (referred to as positive pairs) are closely grouped, while dissimilar ones (referred to as negative pairs) are positioned farther apart [18–20]. This approach has been productively utilized across a spectrum of vision tasks, such as text-to-image generation [21], image-to-image translation [22], image classification [23], as well as natural language comprehension [24]. Of note is the successful application of contrastive learning to single object tracking [25, 26] and multiple object tracking [27, 28]. Nonetheless, these applications typically necessitate additional annotations for positive pairs collection, which wastes ample amount of time in addition to significantly increasing computing complexity [26]. Alternatively, contrastive learning in these approaches is intricately linked with complex and resource-intensive tracking frameworks [25, 27, 28], this scenario makes the transfer of the learning mechanism to UAV tracking unfeasible. Building on these insights, this paper aims to integrate contrastive learning into UAV tracking by adopting an approach that is both efficient and streamlined. This method not only eradicates the necessity for manual annotations but also enables the development and deployment of a lightweight model.

In this work, we utilize intra- and inter-video target templates as contrastive instances to enhance discriminative representation learning for UAV tracking. Unlike traditional contrastive learning methods [23] that generate positive pairs through image augmentation, we generate positive pairs from video data. To address challenges in selecting positive samples, such as occluded targets, we empirically choose two frames randomly from the video to create positive sample pairs, as we have observed that the majority of these pairs demonstrate satisfactory quality. As a result, the proposed tracker adopts the discriminative representations with contrastive instances (DRCI) approach, achieving unparalleled efficiency and precision compared to existing CPU-based and lightweight DL-based trackers in the field of UAV tracking. The acquired discriminative representations serve as a focal point in empowering the model to discern and prioritize crucial features during tracking. By emphasizing essential characteristics, the model becomes adept at filtering out irrelevant variations, thereby reducing the impact of factors like scale, pose, or illumination that may otherwise hinder accurate tracking. This intrinsic capacity to concentrate on pertinent information not only fortifies the model's resilience but also ensures its adaptability across diverse conditions. In summary, the proposed discriminative representation learning with contrastive learning equips our UAV tracking model with the ability to discern and concentrate on key features, paving the way for robust and effective tracking across a spectrum of challenging scenarios. Importantly, our DRCI model does not introduce additional computational load during the inference phase, ensuring consistent performance. To further improve efficiency, we introduced Nvidia TensorRT to quantify the model on the original basis. And have shown the specific performance in Fig. 1. The main contributions of this paper are summarized as follows:

- We lead the vanguard in exploring contrastive learning for UAV tracking, introducing a pioneering feature-learning perspective that yields lightweight DL-based trackers.
- We introduce the DRCI tracker, which acquires discriminative representations through contrastive instances, successfully striking a notable equilibrium between tracking efficiency and precision.
- Our approach is substantiated through application to four prominent public UAV benchmarks, where experimental findings underscore the DRCI tracker's achievement of state-of-the-art performance.

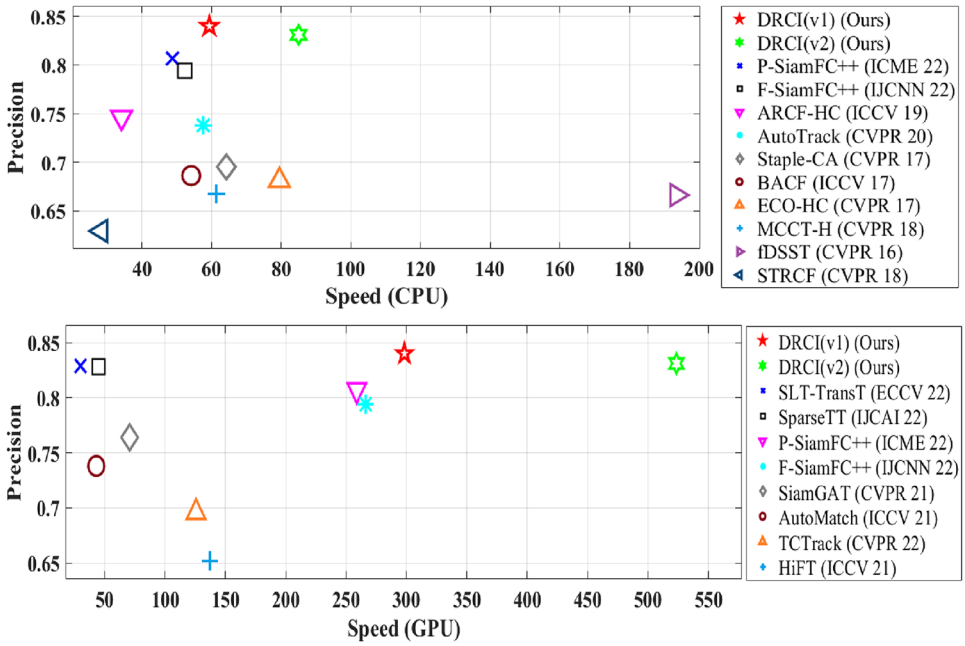

Fig. 1 Using UAVDT [29] data set as the testing benchmark, compared with trackers based on DCF and DL, our DRCI tracker achieves the best balance between precision and efficiency with only a single CPU and GPU in the UAV benchmarks. DRCI(v1) is an improvement of P-SiamFC++ [3] and DRCI(v2) is just a TensorRT conversion of DRCI(v1)

2 Related work

In this paper, we have enhanced and expanded upon our prior research [30] with a thorough examination of how to utilize contrastive instances to learn more discriminative representations for real-time UAV tracking. We have incorporated model quantification methodologies to augment the efficacy of the previous version, thereby achieving a superior balance between the velocity of tracking and its accuracy. This progressive innovation has significantly notched up our real-time tracking performance, clocking an extraordinary average of over 85.4 FPS on a single CPU. Surprisingly, the GPU speed has reached 558.2 FPS. To avoid confusion, the original version is indicated as DRCI(v1), and the upgraded, more advanced version is denoted as DRCI(v2).

2.1 UAV tracking methods

In the expansive realm of contemporary visual tracking techniques, tracker systems broadly classify into two principal categories: DCF-based trackers and DL-based trackers. DCF-based trackers have gained significant traction in UAV tracking due to their noteworthy efficiency. These trackers find their roots in the genesis of the minimum output sum of squared error (MOSSE) filter, marking a pivotal starting point in their evolution. Over time, these trackers have experienced substantial advancements through numerous iterations and diverse variants [7, 31], solidifying their status as cutting-edge methodologies in the domain of UAV tracking. [1, 10, 32–34]. Although DCF-based trackers offer notable advantages, such as increased efficiency, they struggle to maintain robustness, particularly in challenging conditions. This limitation primarily originates from the suboptimal representation capability of handcrafted features. These manually designed features often struggle to consistently capture the nuances inherent in complex tracking scenarios, leading to a shortfall in their adaptability and performance reliability across varied and challenging visual contexts.

In recent years, the realm of visual tracking has experienced substantial advancements propelled by deep learning techniques, notably elevating tracking precision and robustness. This progress is particularly noticeable in the creation of specialized DL-based trackers for UAV applications, which demonstrate significant improvements. For instance, Cao et al. [2] pioneered a hierarchical feature transformer, facilitating the fusion of spatial information and semantic cues to enrich tracking capabilities. Similarly, Fu et al. [35] introduced a two-stage Siamese network-based approach adept at generating and refining high-quality anchor proposals, further bolstering tracking accuracy. In addition, Cao et al. [36] innovatively presented a framework that maximizes temporal context utilization through an adaptive temporal transformer, specifically designed for aerial tracking scenarios. These innovations collectively represent a significant stride in harnessing deep learning for enhancing UAV-based visual tracking methodologies. Despite significant strides, the efficiency of DL-based trackers still falls short compared to many DCF-based counterparts. Recent research aimed to augment the efficiency of DL-based UAV trackers by concentrating on leveraging model compression techniques, as highlighted in studies, such as [3, 4]. Nevertheless, despite the simplicity of these approaches, attaining satisfactory tracking precision at higher compression rates remains a challenging endeavor. Moreover,

despite the emergence of Aba-ViTrack [37], leveraging vision transformer models (ViTs) and dynamically discarding tokens using learned halting probabilities, its efficiency in UAV tracking remains notably inferior when compared to DL-based trackers. This disparity underscores the ongoing challenges in optimizing DL-based methodologies for UAV-specific tracking tasks. In contrast, our paper diverges from conventional methods by presenting a distinctive approach to tackle the issue of subpar performance in UAV tracking. We delve into the realm of contrastive learning, proposing it as a novel technique for feature learning. Our primary aim is to improve the discriminative potential of feature representations, with the goal of significantly boosting tracking performance. Leveraging the prowess of contrastive learning, we aim to bolster feature representation, foreseeing a notable boost in tracking capabilities through this innovative approach.

2.2 Contrastive learning

Absolutely, contrastive learning functions by discerning similarities and discrepancies among samples, operating within the representation space. Its core objective revolves around consolidating similar samples closer together while concurrently pushing dissimilar ones apart, effectively enhancing the representation's discriminative capabilities. Owing to its impressive performance in self-supervised learning paradigms, contrastive learning has garnered significant attention within the field, emerging as a central hub for innovative advancements [21–24]. Contrastive learning has demonstrated its versatility across various fields, including multiple object tracking [27, 28] and single object tracking [25, 26], albeit relatively recently. For example, Pang et al. [27] introduced the concept of quasi-dense similarity learning, which involves densely sampling region proposals from image pairs. This innovative approach enables contrastive learning to capitalize on the most informative regions, thereby enhancing the effectiveness of the learning process. Similarly, Yu et al. [28] devised a trajectory-level contrastive loss strategy, leveraging inter-frame information within target trajectories. This strategy capitalizes on temporal relationships to enhance representations, leading to improved tracking precision and increased robustness. In another vein, Wu et al. [25] presented a progressive unsupervised learning (PUL) framework tailored explicitly to distinguish objects from backgrounds. This framework adopts a progressive learning strategy to refine representations, allowing for more nuanced differentiation between objects and their surroundings, ultimately enhancing tracking accuracy. Similarly, Pi et al. [26] employed contrastive learning to construct instance-aware and category-aware modules. By leveraging different semantic levels, this innovative approach facilitates the creation of robust feature embeddings, enhancing the system's ability to discern both specific instances and broader categories within tracking contexts.

Indeed, a common challenge with these approaches involves the need for extra annotations to gather positive pairs, a process that can be both costly and time-consuming [26]. Moreover, the integration of contrastive learning within these methodologies tends to be intricately connected with complex and resource-demanding tracking frameworks [25, 27, 28]. This interdependency poses a significant hurdle in directly applying these learning mechanisms to the realm of UAV tracking, where resource constraints and real-time processing requirements present considerable challenges. In our paper, our primary goal is to streamline the integration of contrastive learning to yield more discriminative feature representations. Our approach targets the augmentation of both accuracy and efficiency within lightweight DL-based trackers, specifically tailored for UAV tracking. Importantly, we aim to achieve these enhancements without entailing the complexities linked to extensive annotations or the resource-intensive nature of existing frameworks. This streamlined approach seeks to render contrastive learning more accessible and practical within the domain of UAV tracking, ensuring a more efficient and effective tracking system.

3 Methodology

3.1 Overview

In Fig. 2, the proposed discriminative representation using contrastive instances (DRCI) architecture is composed of distinct components: a backbone, a neck, a head network, and a discriminative representation learning (DRL) module. The backbone network, represented as $\phi(\cdot)$, functions as a Siamese network shared between the template and search branches. It processes the template image $\mathbf{Z}$ and the search image $\mathbf{X}$ as inputs, respectively. The neck section integrates four convolutional layers designed to manipulate feature sizes. Following the neck, the head comprises two dense branches, succeeded by three convolutional layers. These layers produce outputs for classification, quality assessment, and regression tasks. The backbone features from both branches undergo size adjustments within the neck before being merged through cross-correlation. The resulting coupled features are then fed into the classification and regression heads for further processing. The coupling of features is formulated as follows:

$$f_l(\mathbf{Z}, \mathbf{X}) = E_2(\psi_l^z(\phi(\mathbf{Z}))) \star E_2(\psi_l^x(\phi(\mathbf{X}))), l \in \{cls, reg\}, \tag{1}$$

the cross-correlation operation utilized in the architecture is symbolized by $\star$, and E_2 represents the encoder responsible for identity-linked feature embedding. Within the

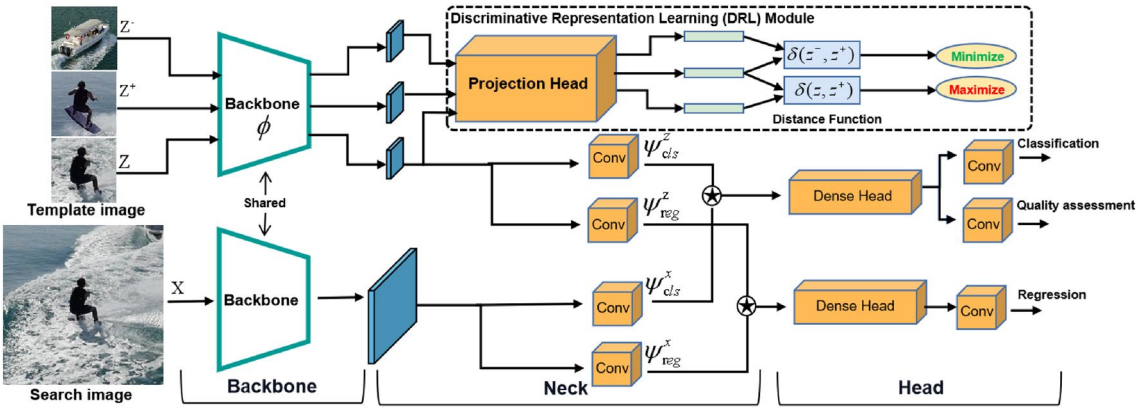

Fig. 2 In this illustration of DRCI, $\psi^{\cdot}_{cls}$ and $\psi^{\cdot}_{reg}$ represent task-specific convolutional layers dedicated to classification and regression tasks, respectively. The template Z functions as an anchor within our contrastive learning setup. In addition, we utilize Z^+ and Z^- as positive and negative samples, respectively, in our contrastive learning framework. These elements collectively form the basis of our DRCI method, leveraging contrastive learning to refine feature representations for improved classification and regression performance

framework, specific layers tailored for classification and regression tasks are denoted as $\psi^x_{cls}(\cdot)$ and $\psi^x_{reg}(\cdot)$, respectively, sharing identical output sizes. Correspondingly, another set of task-specific layers, indicated as $\psi^z_{cls}(\cdot)$ and $\psi^z_{reg}(\cdot)$, serve analogous roles. Throughout the training phase, a DRL module is integrated to bolster the discriminative potential of feature representations specifically tailored for UAV tracking. However, during the inference stage, the DRL module is omitted, thereby eliminating any additional computational overhead in the implementation of our DRCI. We would like to note that in this work we employ blockwise pruning ratios rather than layerwise ratios as used in P-SiamFC++ [3]. Our approach simplifies the pruning process and the search for optimal or sub-optimal pruning ratios, especially considering that determining layerwise pruning ratios in P-SiamFC++ is a tedious and time-consuming task. In addition, we also use quantization techniques to improve efficiency, which is not explored by P-SiamFC++.

3.2 Discriminative representation learning (DRL)

The discriminative representation learning (DRL) module in our UAV tracking model employs a contrastive learning framework, leveraging positive and negative instance pairs during training, which encourages the model to pull together representations of similar instances (positives) while pushing apart those of dissimilar instances (negatives). The learned discriminative representations with DRL enable the model to concentrate on essential features, reducing the influence of irrelevant variations during tracking. This enhances the model's robustness against changes like scale, pose, or illumination, ensuring effective tracking in diverse conditions.

The DRL module integrates a projection head, denoted as $Proj(\cdot)$, to transform the backbone features into an embedding space. This process aims to effectively assess similarity using a relatively straightforward distance function. For simplicity, we adopt a projection head instantiation consisting of a fully connected layer followed by a ReLU activation, akin to the design in SimCLR [23]. While a more sophisticated design for the projection head could potentially lead to performance enhancements, exploring such improvements is a direction we leave for future research endeavors. In obtaining instance samples for contrastive learning, we initially create a minibatch comprising N frame pairs extracted from N distinct sequences. From these pairs, we extract target templates, resulting in N positive pairs and $(C_N^2 - N)$ negative contrastive pairs. These contrastive template samples are represented as $\{\mathbf{Z}_i\}_{i=1}^{2N}$. Here, let $I \equiv \{1, ..., 2N\}$, and $j(i)$ denotes the index of the other sample from the same target, forming a positive pair indicated by $\mathbf{Z}_i \leftrightarrow \mathbf{Z}_{j(i)}$.

For our discriminative representation learning, we adopt the supervised contrastive loss proposed in [38]. Please note that in SimCLR [23], true positive sample pairs are not available, a characteristic referred to as self-supervised contrastive learning in [38]. However, when positive sample pairs are available, it is referred to as supervised contrastive learning in the same work, regardless of whether the negative samples are true or pseudo. The loss takes the following form:

$$L_{DRL} = \sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i)} log \frac{exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} exp(z_i \cdot z_a / \tau)}, \tag{2}$$

given $z_i = Proj(\phi(\mathbf{Z}_i))$, where $\cdot$ denotes the inner product and $\tau \in \mathbb{R}^+$ represents a scalar temperature parameter. In this context, the function $A(i) = I \setminus \{i\}$ defines the set of

indices excluding i. In addition, $P(i) = \{p \in A(i) : \mathbf{Z}_p \leftrightarrow \mathbf{Z}_i\}$ denotes the indices of all positive samples in the minibatch for i, excluding itself. The notation $|P(i)|$ signifies the cardinality of $P(i)$. The primary objective of the DRL loss is twofold: first, to strengthen the similarity between feature representations of targets within the same sequence and second, to reduce the similarity between those originating from different sequences. This serves as the guiding principle behind optimizing the contrastive learning process within the DRL framework.

3.3 Classification, regression and quality assessment losses

The classification branch is tasked with predicting the category at each location, while the regression branch computes the target bounding box for that specific location. The outputs from these branches are represented as $\mathbf{O}^{cls}_{h\times w\times 2}$ and $\mathbf{O}^{reg}_{h\times w\times 4}$, where w and h denote the width and height, respectively. In detail, $\mathbf{O}^{cls}_{h\times w\times 2}(i,j,:)$ signifies a 2D vector portraying the foreground and background scores at the position (i, j). On the other hand, $\mathbf{O}^{reg}_{h\times w\times 4}(i,j,:)$ represents a 4D vector illustrating the distances from the corresponding position to the four sides of the bounding box. Simultaneously, the quality assessment branch functions in parallel with the classification branch and generates an output designated as $\mathbf{O}^{qs}_{h\times w\times 1}$. This output serves to evaluate the quality of classification, subsequently impacting the reweighting of the classification score. Following the approach in P-SiamFC++ [3], the losses for learning these tasks are as follows:

$$L_{CRQ} = \frac{1}{N_{pos}} \sum_z (L_{cls}(p_z, p_z^*) + \lambda_1 I_{\{p_z^*>0\}} L_{reg}(t_z, t_z^*) + \lambda_2 I_{\{p_z^*>0\}} L_{qs}(q_z, q_z^*)) \tag{3}$$

The comprehensive loss function for training our DRCI combines three primary components: the focal loss (L_{cls}), the IoU loss (L_{reg}), and the binary cross-entropy loss (L_{qs}). These components correspond to the tasks of classification, regression, and quality assessment, respectively. In these expressions, the symbol z denotes a coordinate on a feature map, p_z represents a prediction, and p_z^* signifies the corresponding target label. The function $I_{\{\cdot\}}$ serves as the indicator function, aiding in calculations, while $N_{pos} = \sum_z I_{\{p_z^*>0\}}$ signifies the count of positive samples. The weight parameters λ_1 and λ_2 are introduced to balance these individual losses within the overall framework. It is important to note that in the classification task, p_z^* is assigned the value of 1 if z is identified as a positive sample and 0 otherwise, delineating the positive and negative samples for this specific task.

In summary, the overall loss for training our DRCI can be expressed as

$$L = L_{CRQ} + \rho L_{DRL}, \tag{4}$$

where ρ is a constant coefficient to balance L_{CRQ} and L_{DRL}. We emphasize that the rationale for combining L_{CRQ} and L_{DRL} stems from the complementary nature of these loss functions and their potential to enhance the overall performance and robustness of the model. This approach not only simplifies the training process but also encourages the model to learn more robust and generalized representations that can improve performance across various tracking scenarios.

4 Experiments

In this section, we present a thorough evaluation of our proposed DRCI tracker, assessing both DRCI (v1) and DRCI (v2) for their superior performance and robustness. Our evaluation encompasses four widely acknowledged benchmarks for aerial tracking: UAVDT [29], DTB70 [44], VisDrone2018 [45] and UAV123@10fps [46]. These benchmarks hold significant recognition within the field and are extensively utilized for evaluating UAV tracker performance. Code will be available on: https://github.com/P-SiamFCpp/DRCI.

4.1 Evaluation data set

UAVDT [29], showcases a diverse array of complex scenarios captured by drones. Its primary focus lies in vehicle tracking across different weather conditions, flight altitudes, and camera perspectives. This data set serves as a robust evaluation platform for tracking algorithms in real-world UAV scenarios. DTB70 [44], encompasses a collection of 70 sequences captured by drones. This data set encompasses both short-term and long-term aerial targets and is characterized by chaotic scenes involving objects of varying sizes. DTB70 is specifically curated to challenge tracking algorithms in diverse and challenging UAV scenarios, providing a comprehensive evaluation environment. VisDrone2018 [45], stands out as a substantial and expansive data set crafted for drone-based vision applications. This data set offers a wide spectrum of scene and object categories, comprising high-resolution images, videos, annotations, and metadata. It serves as a rich resource for evaluating tracking algorithms within the domain of UAV vision, presenting diverse scenarios and extensive annotations for comprehensive assessment. UAV123@10fps [46], this data set is specifically created by subsampling UAV123 the original 30FPS (frames per second) data set to a lower capture

rate of 10FPS. The primary aim of UAV123@10fps is to investigate and analyze the impact of varying camera capture rates on the performance of tracking algorithms. This data set provides valuable insights into how frame rate alterations affect the efficacy of UAV tracking method.

4.2 Experimental setup

The evaluation experiments were conducted on a system running Ubuntu 18.04, equipped with an NVIDIA TitanX GPU, an i9-10850K processor (3.6GHz), and 16GB RAM. The system utilized Nvidia-Driver-470 and CUDA version 10.1. It should be noted that variations in device configurations might yield slight differences in experimental results. Both versions of the DRCI tracker, namely, DRCI (v1) and DRCI (v2), were pruned using blockwise ratios: 0.7 for the backbone and 0.5 for the neck and 0.3 for the head, as detailed in [5]. The remaining architecture components follow the structure of F-SiamFC++. The temperature parameter (τ) was set to 0.5, and the default setting for ρ was 0.1. In addition, other parameters crucial for training and inference, such as λ_1 and λ_2, were adopted from the P-SiamFC++ framework for consistency and comparability in the evaluation process. We use the GOT-10k [47] training data set with a batch size of 32 and 8 workers. Initially, a 5-epoch warm-up phase linearly increases the learning rate from 10^{-7} to 2×10^{-3}. Subsequently, a cosine annealing learning rate schedule is adopted for the remaining 15 epochs. Each epoch involves processing 600k image pairs. SGD optimization with a momentum of 0.9 is employed throughout. The backbone of DRCI (v2) is accelerated with Nvidia TensorRT, quantifying it from float 32 to float 16 bit for enhancement. The purpose is to improve the speed of CPU and GPU through quantification. Specifically, in DRCI (v2), it is necessary to perform quantitative acceleration processing on the network backbone module and it is necessary to convert the trained model into TensorRT format before acceleration, then use the TensorRT inference engine to run this model, while others are the same as in DRCI (v1) version.

4.3 Comparison with CPU-based trackers

Eight state-of-the-art (SOTA) trackers based on hand-crafted features for comparison are: RACF [7], AutoTrack [1], BACF [42], ARCF-HC [10], STRCF [43], ECO-HC [41], fDSST [40], KCF [39]. Figure 3 visually represents the precision and success rate achieved across four key UAV benchmark tests, specifically UAV123@10fps [46], DTB70 [44], UAVDT [29], and VisDrone2018 [45], arranged in sequence from left to right. Similarly, Fig. 4 showcases the evaluation outcomes related to partial attributes observed in the corresponding UAV benchmark tests. Furthermore, Table 1 provides a comprehensive overview, presenting the average performance metrics, including precision (PRC) and frames per second (FPS) attained on a single CPU.

Overall performance evaluation: The performance comparison of DRCI against competing trackers across the four UAV benchmarks is presented in Fig. 3. Notably, both DRCI (v1) and DRCI (v2) exhibit superior performance over all other trackers across most benchmarks, with the exception being VisDrone2018. Particularly, concerning UAV123@10fps, DTB70, and UAVDT, DRCI (v1) showcases significant superiority over the 2nd ranked tracker RACF in terms of precision and success rates, achieving noteworthy gains of (4.2%, 6.6%), (8.9%, 11.3%) and (6.7%, 9.6%), respectively. And DRCI (v2) is batter than it with (4.6%, 7.0%), (9.1%, 11.3%) and (5.8%, 8.7%). It can be seen that the performance difference between DRCI (v1)

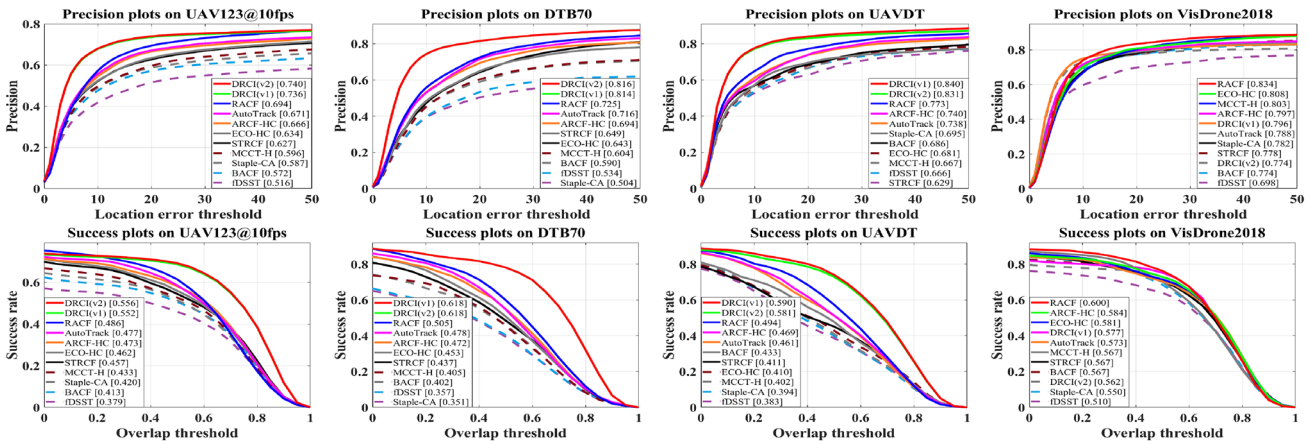

Fig. 3 Overall performance of hand-crafted trackers on various data sets, including UAVDT, DTB70, VisDrone2018 and UAV123@10fps, is evaluated using precision and success rate for one-pass evaluation (OPE). Precision at 20 pixels and the area under the curve (AUC) are employed as evaluation metrics, and they are indicated on the precision plots and success plots, respectively, to determine the ranking. The performance is assessed from left to right on the mentioned data sets

Table 1 Average precision and speed (FPS) of DRCI and hand-crafted-based trackers were compared on UAVDT, DTB70, VisDrone2018 and UAV123@10fps. It is worth noting that all reported FPS values were evaluated on a single CPU. The precision values are differentiated by using the colors red, blue and green, which respectively indicate the first, second, and third places among the rankings of the tracking methods

	KCF [39]	fDSST [40]	ECO-HC [41]	BACF [42]	ARCF-HC [10]	AutoTrack [1]	STRCF [43]	RACF [7]	**DRCI (v1)**	**DRCI (v2)**
Precision	53.3	60.4	68.8	64.2	71.9	72.3	67.1	**75.7**	**79.7**	**79.1**
FPS (CPU)	**622.5**	**193.4**	84.5	54.2	34.2	58.7	28.4	35.7	58.9	**85.4**

and DRCI (v2) in these three UAV benchmarks is not significant. Even DRCI (v2) performs better than DRCI (v1) on the two UAV benchmarks. On VisDrone2018, our DRCI (v2) is inferior to the first tracker RACF in precision and success, the gaps are 6.0% and 3.8%, the gap between DRCI (v1) and it are 3.8% and 2.3%, respectively. This variance in performance can be attributed to RACF's parameter optimization specifically tailored to the characteristics of particular data sets. In contrast, our DRCI doesn't depend on data set-specific parameter tuning, potentially resulting in slightly lower performance compared to methods fine-tuned for data set-specific nuances. In terms of precision, DRCI (v1) also slightly outperforms MCCT-H, ARCF-HC, and ECO-HC, with the maximum gap being 1.1%. And it is surpass ECO-HC and ARCF-HC in terms of success, with the maximum gap being 0.7%. Even if model quantification methodologies is used on DRCI (v2), performed slightly worse than other methods only on this UAV benchmarks, but outperformed all compared methods on the other three UAV benchmarks, even higher than DRCI (v1). This alone indicates that model quantization is successful in UAV tracking, as the speed improvement of DRCI (v2) on a single CPU is even more astonishing. Regarding speed, we evaluate tracking performance using the average Frames Per Second (FPS) across the four UAV benchmarks mentioned. Table 1 presents the average precision and FPS results for various trackers. It is evident that both DRCI (v1) and DRCI (v2) outperform all other competing trackers in terms of precision and stand out as the top-performing real-time trackers with a speed exceeding 30 FPS on CPU. Our DRCI (v1) has achieved an impressive precision rate of 79.7% while maintaining a high speed of 58.9 frames per second (FPS). Specifically, although DRCI (v2) achieved the precision of 79.1%, which is 0.6% lower than DRCI (v1), it achieved 85.4 FPS on CPU speed, which is 45.0% higher than DRCI (v1). In addition, it is three times that of the best real-time tracker (speed of >30FPS) on CPU. Taking the reduced precision as a percentage, DRCI (v2) is only 0.7% lower than DRCI (v1). Compared to the 45.0% increase on CPU speed, this precision loss can be negligible. Therefore, the proposed DRCI (v2) achieves a better balance between efficiency and precision, which DRCI (v1) cannot achieve.

Attribute-based evaluation: Our DRCI (v1) and DRCI (v2) outperform other competing DCF-based trackers in most of the attributes defined in each of the four benchmark tests. Examples of success plots are shown in Fig. 4. To demonstrate that our method is universal with strong robustness and achieve a balance of precision and efficiency in the vast majority of cases, we present 8 different attributes across the four UAV benchmarks. As can be seen, in the situations of Illumination variation and Viewpoint chang on UAV123@10fps [46], Deformation and Scale variatio on DTB70 [44], Object blur and Small object on UAVDT

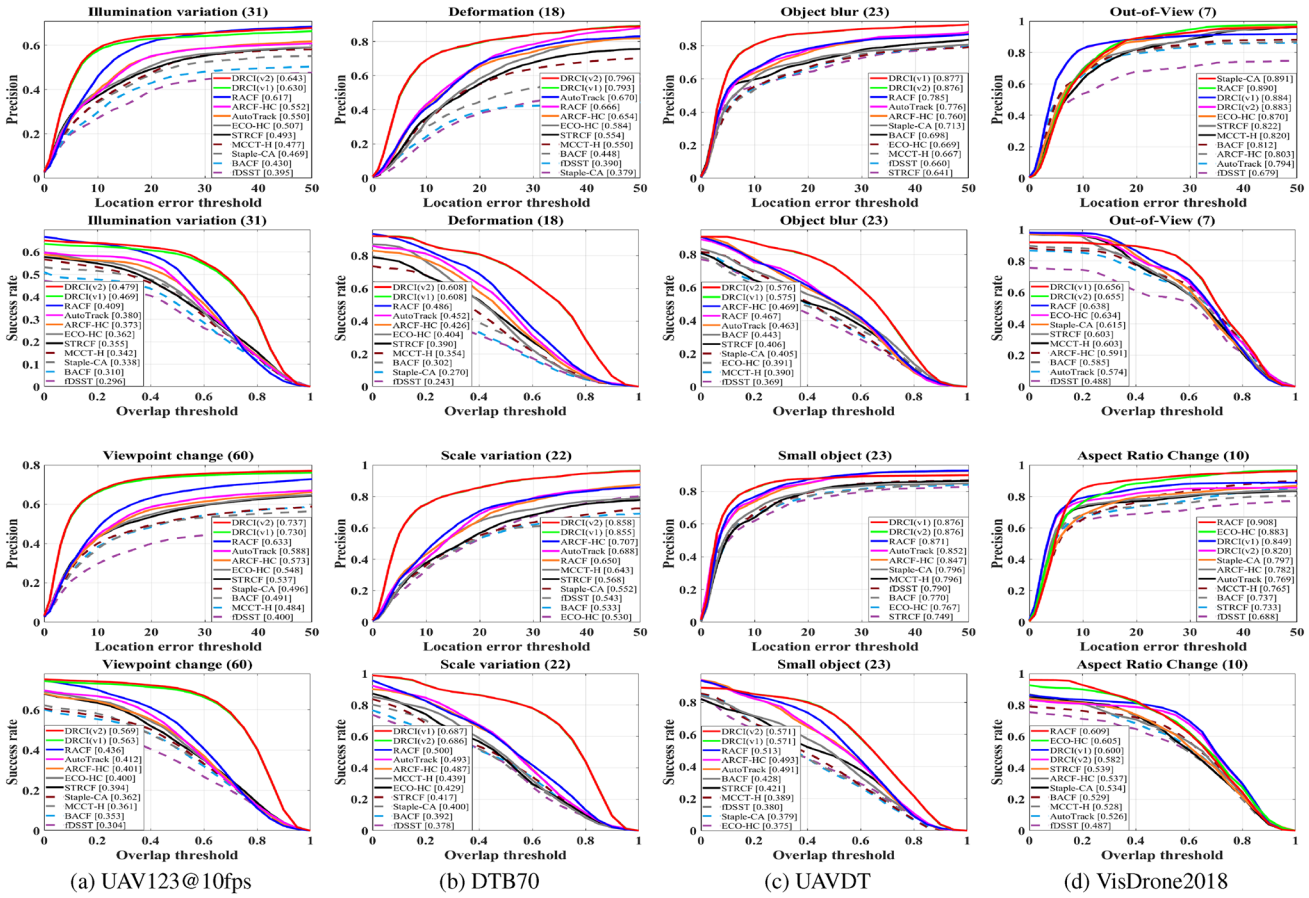

Fig. 4 Attribute-based comparison on illumination variation, viewpoint change, deformation, scale variation, object blur, small object, out-of-view and aspect ratio change. From a to d represent UAV123@10fps [46] DTB70 [44], UAVDT [29] and VisDrone2018 [45], respectively. And each column shows the PRC and AUC of two different attributes of one data set

[29], Out-of-view and aspect ratio change on VisDrone2018 [45], DRCI (v1) and DRCI (v2) demonstrate significant improvements over other trackers because the effectiveness of feature representation with deep learning, justifying the effectiveness of developing lightweight deeper trackers for UAV tracking. It can be clearly observed from the figure that DRCI (v2) and DRCI (v1) have nearly identical performance, especially in the UAV123@10fps and DTB70 UAV benchmarks. Although there is a gap between UAVDT and VisDrone2018 UAV benchmarks, overall, our DRCI (v1) and DRCI (v2) are very close, which can be said to achieve a balance between precision and speed. Specifically, in the UAV123@10fps [46] among the Viewpoint change attributes, DRCI (v2) has the 10.4% and 13.3% higher precision and success rate than the second best RACF, respectively. The precision and success rate of DRCI (v2) even better than the second method with 15.1% and 18.6%, in Scale variation attributes of the DTB70 [44]. This further proves that our DRCI (v1) and DRCI (v2) has excellent performance in improving scale and viewpoint changes. In addition, although other attributes do not have as much improvement as these two attributes, there are also quite better improvements in other specific attributes. The effect on VisDrone2018 [45] is still not ideal, but compared to Fig. 3, there has been a significant improvement, with the vast majority of differences only around 1.0% compared to the best method RACF [7].

4.4 Comparison with DL-based trackers

On the UAVDT [29] and DTB70 [44] data set the proposed DRCI (v1) and DRCI (v2) is also compared with twelve state-of-the-art (SOTA) DL-based trackers, including AutoMatch [54], SparseTT [51], HiFT [2], SLT-TransT [50], TCTrack [36], P-SiamFC++ [3], SiamGAT [52], F-SiamFC++ [4], DropTrack [49], SeqTrack [48], MAT [53], Aba-ViTrack [37]. The table in Table 2 displays the FPS and precision results on UAVDT and DTB70. It is evident that our DRCI (v1) exhibits superior precision and GPU speed compared to competing DL-based trackers. Specifically, it outperforms the second-ranked tracker, Aba-ViTrack [37], by a margin of 0.7% in precision, while

Table 2 Comparative analysis of precision and speed (measured in frames per second—FPS) between DRCI and deep-based trackers specifically on the UAVDT [29] and DTB70 [44] data set. Red, blue and green indicate the first, second and third place

Tracker	UAVDT		Tracker	DTB70	
	PRC	FPS		PRC	FPS
DRCI(v1)	**84.0**	**298.3**	DRCI(v1)	**81.4**	**297.7**
DRCI(v2)	**83.1**	**536.9**	DRCI(v2)	**81.6**	**578.2**
SeqTrack [48]	79.0	13.2	F-SiamFC++ [4]	**81.3**	**250.4**
DropTrack [49]	77.2	23.6	P-SiamFC++ [3]	80.3	238.2
SLT-TransT [50]	82.9	29.9	TCTrack [36]	81.2	128.0
SparseTT [51]	82.8	45.1	HiFT [2]	77.8	133.5
Aba-ViTrack [37]	**83.3**	**175.2**	SiamGAT [52]	79.7	72.1
MAT [53]	72.9	71.2	AutoMatch [54]	71.6	42.8

achieving a GPU speed that is more than 70.3% faster than Aba-ViTrack on UAVDT. Although our DRCI (v2) precision is 0.9% lower than DRCI (v1), it has reached 536.9 FPS on GPU speed, which is 79.9% higher than DRCI (v1). Although slightly inferior in precision to DRCI (v1), it can still rank third. Although 0.2% lower than Aba-ViTrack who ranks second in precision, it is 3 times faster than Aba-ViTrack on GPU speed. On DTB70, our DRCI (v2) ranks first, with a precision 0.2% higher than the second-ranked DRCI (v1), but is almost twice as fast in speed. The outcomes not only affirm the capability of our proposed method to develop a lightweight DL-based tracker with notably superior tracking precision and efficiency but also validate our unique approach to tackling the challenge of low performance in UAV tracking through a novel feature-learning perspective. This strategy effectively amplifies the discriminative potential of feature representations, contributing to enhanced tracking capabilities. Moreover, the substantial surge in speed observed on the GPU serves as a testament to the success of our utilization of quantization techniques in UAV tracking. With quantization, the precision of numerical representations in network parameters and activations is reduced, typically from floating point to lower precision fixed point. This reduction in precision can reduce memory footprint, reduce computational requirements, and speed up inference times. This achievement highlights the effectiveness and success of our strategies in optimizing computational efficiency while maintaining tracking accuracy, signifying a notable advancement in UAV tracking methodologies.

4.5 Qualitative comparison with SOTA trackers

We are displaying qualitative tracking results of our method alongside six state-of-the-art trackers, i.e., TCTrack [36], HiFT [2], RACF [7], AutoTrack [54], ARCF-HC [10] and ECO-HC [41] in Fig. 5. We mainly select two video sequences from each of the four UAV benchmarks, which are person1_s and wakeboard6 of UAV123@10fps, Basketball and BMX4 of DTB70, S0309 and S0310 of UAVDT, uav0000074_01656_s and uav0000164_00000_s of Vistrone2018 from top to bottom. It is evident that among all trackers, only our DRCI (v1) and DRCI (v2) consistently succeed in tracking targets across all eight challenging examples. These scenarios involve objects encountering illumination changes and severe shadow occlusion (i.e., persion1_s and BMX4), as well as pose variations (i.e., BMX4 and S0309). The capability of our trackers to maintain successful tracking in these diverse and challenging scenarios highlights their robustness and adaptability in handling complex real-world conditions. In scenarios involving ultra-long distance tracking of small targets and dealing with challenges like water surface reflection due to sunlight (i.e., wakeboard6), our DRCI (v1) and DRCI (v2) exhibit consistent and accurate tracking throughout. Particularly in the final frame, where other methods lose track, only our method maintains precise tracking. To further underscore the robustness of DRCI, we deliberately selected a video sequence showcasing low illumination in dark settings and multiple intertwined fluorescent lamps affecting each other, presenting challenges in tracking a blurred target. Our method exhibits a remarkable ability to accurately track the target under such challenging conditions, surpassing the capabilities of other methods in achieving accurate tracking. In scenarios where the tracking target includes similar objects in an ultra-long video sequence or severe noise interference, our tracking performance surpasses that of other methods. Notably, upon closer scrutiny, the distinction between DRCI (v1) and DRCI (v2) appears minimal. In these challenging cases, our method consistently showcases markedly improved performance and yields visually superior results compared to alternative methods. This further reinforces the efficacy of our proposed approach, particularly in learning discriminative representations using contrastive instances within the realm of UAV tracking. These emphasize the robustness and reliability of our method in handling diverse and complex tracking scenarios. In Fig. 6, it is evident that every tracker eventually fails to maintain target tracking. The first case involves a blurred car undergoing substantial pose changes, while the second deals with tracking a fast-moving small UAV. In the third case, the task is to track a single goat amidst a group of others, and the final challenge is tracking a person who becomes fully occluded while in motion. These examples highlight the difficulties posed by factors, such as fast motion, clustered backgrounds, extreme visual angle changes, and severe occlusion. Their outcomes underscore the ongoing challenges in UAV tracking efforts.

4.6 Ablation study

Effect of discriminative representation learning (DRL)

We conducted comparisons between the proposed DRCI (v1) and DRCI (v2) against the baseline P-SiamFC++ across all four UAV benchmarks, evaluating model size, precision, and tracking speed to assess their effectiveness. These comparisons are outlined in Table 3. Notably, the model size of DRCI (v1) has been reduced to 67.4% (≈ 5.05/7.49) of its original size. In addition, there have been

Table 3 Comparison table highlights the differences in model size (measured in parameters), precision (PRC), and tracking speed between the proposed DRCI and the baseline method P-SiamFC++ across four UAV benchmarks

Methods	Parameters	UAV123@10fps			DTB70			UAVDT			VisDrone2018			Avg.		
		PRC	FPS		PRC	FPS		PRC	FPS		PRC	FPS		PRC	FPS	
			CPU	GPU		CPU	GPU		CPU	GPU		CPU	GPU		CPU	GPU
P-SiamFC++	7.49M	73.1	45.1	236.4	80.3	45.6	238.2	80.7	48.8	258.8	**80.9**	45.0	230.5	78.8	46.1	241.0
DRCI (v1)	**5.05M**	73.6	59.2	300.7	81.4	60.1	297.7	**84.0**	59.4	298.3	79.6	57.0	284.6	**79.7**	58.9	295.3
DRCI (v2)	**5.05M**	**74.0**	**86.5**	**560.4**	**81.6**	**83.3**	**578.2**	83.1	**85.0**	**536.9**	77.4	**86.7**	**557.2**	79.1	**85.4**	**558.2**

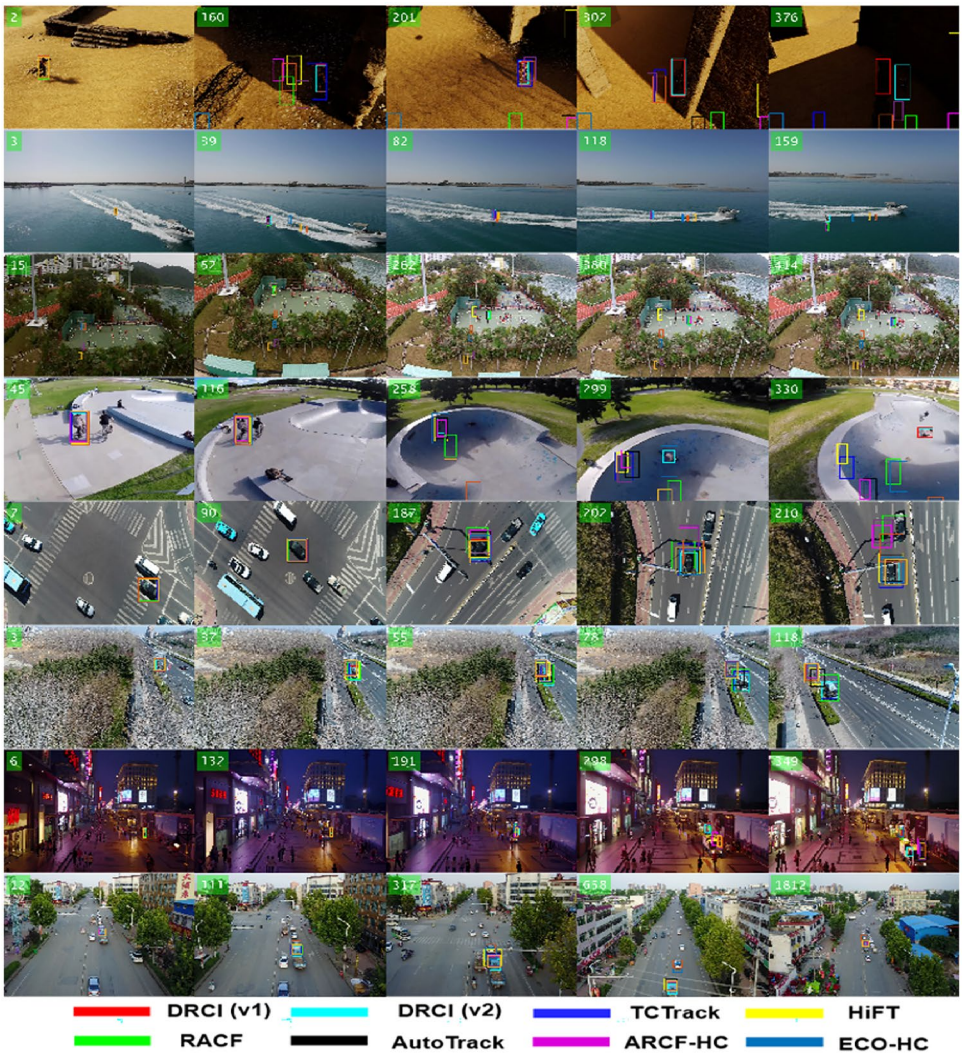

Fig. 5 Qualitative evaluation on 8 sequences from, respectively, UAVDT, UAV123@10fps, DTB70 and VisDrone2018. The different colors represent the different tracking results

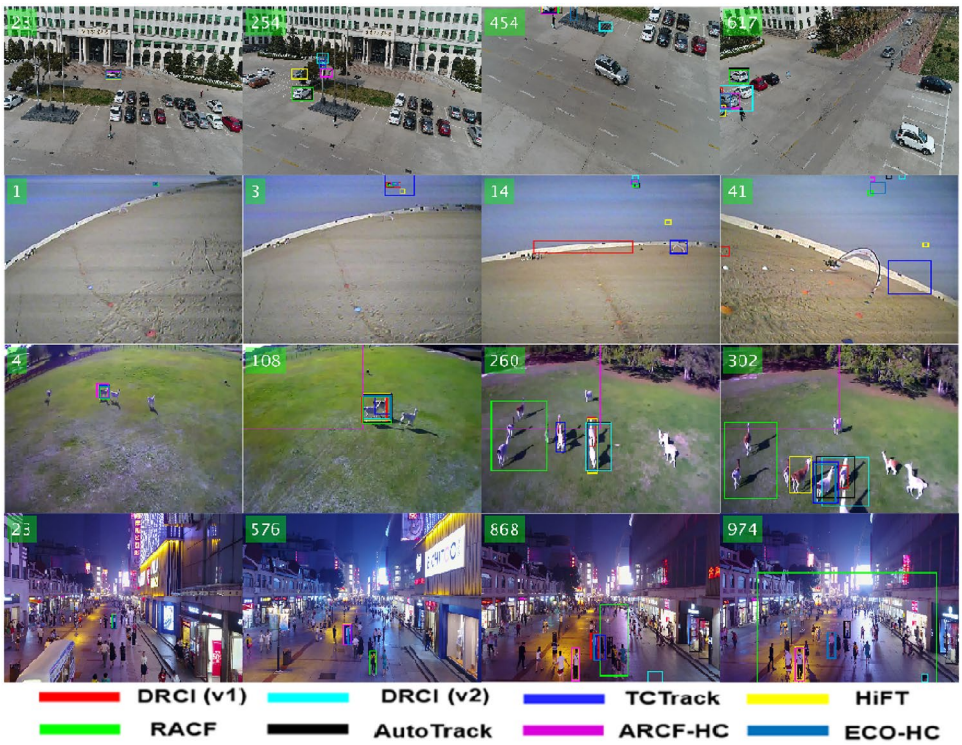

Fig. 6 Qualitative evaluation on 4 sequences from, respectively, UAVDT, UAV123@10fps, DTB70 and VisDrone2018. The different colors represent the different tracking results

improvements in both CPU and GPU speeds. On average, the CPU speed increased from 46.1 to 58.9 FPS, while the GPU speed surged from 241.0 to 295.3 FPS. Despite a marginal 1.3% precision deficit compared to the baseline on the VisDrone2018 data set, DRCI (v1) demonstrates noteworthy improvements on the DTB70 and UAVDT data sets. Particularly, it achieves gains of 1.1% and 3.3% in precision, respectively, on these data sets. These results underscore the strides made by DRCI (v1) in achieving a more compact model size while enhancing tracking speeds and maintaining competitive precision levels across various benchmarks. When compared to the baseline P-SiamFC++, our DRCI (v2) not only demonstrates a superior precision by 0.3% but also achieves a remarkable 85.2% increase in speed on a single CPU. Moreover, it achieves a 1.3 times faster speed than P-SiamFC++ when evaluated in terms of GPU speed. Notably, the improvements in tracking speed for DRCI (v2) compared to DRCI (v1) are striking. On average, the CPU speed elevates from 58.9 FPS to an impressive 85.4 FPS, while the GPU speed surges from 295.3 FPS to a remarkable 558.2 FPS. While slightly trailing behind DRCI (v1) on UAVDT and VisDrone2018 benchmarks, DRCI (v2) surpasses DRCI (v1) on other UAV benchmarks. It is noteworthy that DRCI (v2) achieves a 45.0% higher CPU speed and an 89.0% enhanced GPU speed compared to DRCI (v1). These outcomes strongly support the effectiveness of adopting deep reinforcement learning (DRL) as an innovative feature-learning perspective to enhance UAV tracking. This approach significantly enhances both efficiency and precision, reaffirming its substantial value and impact within the field.

Impact of loss L_{DRL}: To gauge the influence of the DRL (deep reinforcement learning) loss on DRCI's precision, we undertook training iterations using diverse DRL loss weights and subsequently conducted evaluations across four distinct benchmarks. Throughout this evaluation process, we systematically adjusted the weight parameter ρ, referenced in Eq. 4, across a range from 0.0 to 1.0, incrementally increasing by 0.1. This methodical variation enabled a comprehensive exploration of how manipulating the DRL loss weight impacted the precision of the DRCI model across the suite

Table 4 Depicting the fluctuation in DRCI's precision across the four benchmarks concerning the weight parameter (ρ) associated with discriminative representation learning loss

ρ	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DTB70	**80.5**	**81.5**	80.1	78.9	79.0	**80.4**	78.6	78.1	78.9	77.9
UAVDT	76.2	**84.0**	**82.7**	**81.9**	78.9	80.8	81.8	78.9	76.5	79.5
UAV123@10fps	**72.8**	**72.1**	69.9	70.0	69.4	70.8	**71.2**	70.7	69.3	69.5
VisDrone2018	72.5	**79.6**	76.9	**77.4**	76.4	76.0	76.0	74.5	**77.5**	74.5

of benchmarks. Table 4 provides a detailed presentation of the precision outcomes for DRCI across varying values of ρ on four benchmarks. It is important to note that $\rho = 0.0$ corresponds to the performance of the baseline tracker, P-SiamFC++. The results highlight that setting ρ to 0.1 yields the highest precision across four benchmarks, except for UAV123@10fps. Notably, substantial improvements in precision are observed on UAVDT and VisDrone2018 when ρ surpasses 0.0, indicating the beneficial impact of integrating the proposed DRL loss into the DRCI framework. While precision experiences fluctuations on DTB70 and UAV123@10fps, the overall trend emphasizes that the most optimal precisions are achieved when ρ is approximately set to 0.1. This outcome underscores the effectiveness of judiciously applying the proposed DRL loss, substantiating its ability to enhance the baseline tracker's precision and further validates the effectiveness of the proposed DRCI methodology.

5 Conclusion

In this work, we pioneer the exploration of learning discriminative representations using contrastive instances for UAV tracking. This approach not only eliminates the need for manual annotations but also facilitates the development and deployment of lightweight models. Our proposed DRCI has shown its capability to acquire more efficient and compact representations, leading to state-of-the-art performance across four UAV benchmarks in terms of both efficiency and tracking precision. We anticipate that our work will inspire further efforts in the development of more effective and efficient lightweight DL-based trackers for UAV tracking applications. On the other hand, the approach taken to mitigate the substantial reduction in precision involved a compromised quantization process, wherein the model was quantized from float 32 to float 16 bits. As a result, an interesting avenue for future research could focus on exploring more efficient and effective quantization techniques that have the potential to further enhance the model's efficiency without compromising precision.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11554-024-01456-2.

Acknowledgements Thanks to the supports by Guangxi Key Laboratory of Embedded Technology and Intelligent System, Research Institute of Trustworthy Autonomous Systems, the Guangxi Science and Technology Base and Talent Special Project (No. Guike AD22035127), the National Natural Science Foundation of China (No. 62176170, 62066042, 61971005), the Science and Technology Department of Tibet (No. XZ202102YD0018C), the Sichuan Province Key Research and Development Project (No. 2020YJ0282).

References

1. Li, Y., Fu, C., Ding, F., Huang, Z., Lu, G.: Autotrack: towards high-performance visual tracking for uav with automatic spatio-temporal regularization, CVPR, pp. 11920–11929 (2020)
2. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: Hift: hierarchical feature transformer for aerial tracking. In: ICCV, pp. 15457–15466 (2021)
3. Wang, X., Zeng, D., Zhao, Q., Li, S.: Rank-based filter pruning for real-time uav tracking. In: ICME, pp. 01–06 (2022)
4. Wu, W., Pengzhi, Z., Li, S.: Fisher pruning for real-time uav tracking. In: IJCNN, pp. 1–7 (2022)
5. Wang, X., Zeng, D., Zhao, Q., Li, S.: Exploiting rank-based filter pruning for real-time uav tracking. SSRN Electron. J. (2022)
6. Wang, X., Yang, X., Ye, H., Li, S.: Learning disentangled representation with mutual information maximization for real-time uav tracking. ICME, pp. 1331–1336 (2023)
7. Li, S., Liu, Y., Zhao, Q., Feng, Z.: Learning residue-aware correlation filters and refining scale for real-time uav tracking. Pattern Recogn. **127**, 108614 (2022)
8. Gao, P., Zhang, Q., Xiao, L., Zhang, Y.L., Wang, F.: Learning reinforced attentional representation for end-to-end visual tracking. Elsevier, Amsterdam. arXiv:abs/1908.10009 (2019)
9. Gao, P., Ma, Y., Yuan, R., Xiao, L., Wang, F.: Siamese attentional keypoint network for high performance visual tracking. Knowl. Based Syst. **193**, 105448 (2019)
10. Huang, Z., Fu, C., Li, Y., Lin, F., Lu, P.: Learning aberrance repressed correlation filters for real-time uav tracking, ICCV, pp. 2891–2900 (2019)
11. Zhang, Z., Wu, F., Qiu, Y., Liang, J., Li, S.: Tracking small and fast moving objects: A benchmark. In: ACCV (2022)
12. Zhang, Z., Wu, F., Qiu, Y., Liang, J., Li, S.: Tsfmo: a benchmark for tracking small and fast moving objects. SSRN Electron. J. (2023)
13. Gao, P., Zhang, X., Yang, X.-L., Gao, F., Fujita, H., Wang, F.: Robust visual tracking with extreme point graph-guided annotation: approach and experiment. Expert Syst. Appl. **238**, 122013 (2024)
14. Liu, M., Wang, Y., Sun, Q., Li, S.: Global filter pruning with self-attention for real-time uav tracking. In: BMVC (2022)
15. Zhong, P., Zeng, D., Wang, X., Li, S.: Efficiency and precision trade-offs in uav tracking with filter pruning and dynamic channel weighting. In: Fuzzy Systems and Data Mining (2022)
16. Liu, J., Zhuang, B., Zhuang, Z., Guo, Y., Huang, J., Zhu, J.-H., Tan, M.: Discrimination-aware network pruning for deep model compression. IEEE Trans. Pattern Anal. Mach. Intell. **44**, 4035–4051 (2020)
17. Jordão, A., Lie, M., Schwartz, W.R.: Discriminative layer pruning for convolutional neural networks. IEEE J. Select. Top. Signal Process. **14**, 828–837 (2020)
18. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 1798–1828 (2012)
19. Ye, F., Bors, A.: Learning joint latent representations based on information maximization. Inf. Sci. **567**, 216–236 (2021)
20. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. Proc. IEEE **109**, 612–634 (2021)
21. Zhang, H., Koh, J. Y., Baldridge, J., Lee, H., Yang, Y.: Cross-modal contrastive learning for text-to-image generation. CVPR, pp. 833–842 (2021)
22. Park, T., Efros, A. A., Zhang, R., Zhu, J.-Y.: Contrastive learning for unpaired image-to-image translation. In: ECCV (2022)
23. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. arXiv:abs/2002.05709 (2020)

24. Li, S., Hu, X., Lin, L., Wen, L.: Pair-level supervised contrastive learning for natural language inference. ICASSP, pp. 8237–8241 (2022)
25. Wu, J., Wan, A.B.: Chan, CVPR, Progressive unsupervised learning for visual object tracking (2021)
26. Pi, Z., Wan, W., Sun, C., Gao, C., Sang, N., Li, C.: Hierarchical feature embedding for visual tracking. In: ECCV (2022)
27. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. CVPR, pp. 164–173 (2021)
28. Yu, E., Li, Z., Han, S.: Towards discriminative representation: multi-view trajectory contrastive learning for online multi-object tracking. CVPR, pp. 8824–8833 (2022)
29. Du, D., Qi, Y., Yu, H., Yang, Y.-F., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q.: The unmanned aerial vehicle benchmark: Object detection and tracking. In: ECCV, pp. 375–391 (2018)
30. Zeng, D., Zou, M., Wang, X., Li, S.: Towards discriminative representations with contrastive instances for real-time uav tracking. In: ICME, pp. 1349–1354 (2023)
31. Gao, P., Ma, Y., Song, K., Li, C., Wang, F., Xiao, L., Zhang, Y.: High performance visual tracking with circular and structural operators. Knowl. Based Syst. **161**, 240–253 (2018)
32. Li, S., Liu, Y., Zhao, Q., Feng, Z.: Learning residue-aware correlation filters and refining scale estimates with the grabcut for real-time uav tracking. 3DV, pp. 1238–1248 (2021)
33. Li, S.-W., Jiang, Q.-B., Zhao, Q.-J., Lu, L., Feng, Z.-L.: Asymmetric discriminative correlation filters for visual tracking. Front. Inf. Technol. Electron. Eng. **21**(10), 1467–1484 (2020)
34. Li, S., Zhao, Q., Feng, Z., Lu, L.: Equivalence of correlation filter and convolution filter in visual tracking. arXiv:abs/2105.00158 (2021)
35. C. F., Z. C., , et al., Siamese anchor proposal network for high-speed aerial tracking. ICRA, pp. 510–516 (2021)
36. Cao, Z., Huang, Z., Pan, L., Zhang, S., Liu, Z., Fu, C.: Tctrack: Temporal contexts for aerial tracking. CVPR, pp. 14778–14788 (2022)
37. Li, S., Yang, Y., Zeng, D., Wang, X.: Adaptive and background-aware vision transformer for real-time uav tracking. In: ICCV, pp. 13989–14000 (2023)
38. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. arXiv:abs/2004.11362 (2020)
39. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**, 583–596 (2014)
40. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Adaptive decontamination of the training set: a unified formulation for discriminative visual tracking. In: CVPR, pp. 1430–1438 (2016)
41. Danelljan, M., Bhat, G., Khan, F. S., Felsberg, M.: Eco: efficient convolution operators for tracking. In: CVPR, pp. 6931–6939 (2016)
42. Galoogahi, H.K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. ICCV, pp. 1144–1152 (2017)
43. Li, F., Tian, C., Zuo, W., Zhang, L., Yang, M.-H.: Learning spatial-temporal regularized correlation filters for visual tracking. In: CVPR, pp. 4904–4913 (2018)
44. Li, S., Yeung, D.Y.: Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models. In: AAAI (2017)
45. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H., et al.: Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In: ECCV Workshops (2018)
46. : Mueller, M., Smith, N. G., Ghanem, B.: A benchmark and simulator for uav tracking. In: ECCV (2016)
47. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence (5) (2021)
48. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: Seqtrack: sequence to sequence learning for visual object tracking. CVPR, pp. 14572–14581 (2023)
49. Wu, Q., Yang, T., Liu, Z., Wu, B., Shan, Y., Chan, A. B.: Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. CVPR, pp. 14561–14571 (2023)
50. Kim, M., Lee, S., Ok, J. Han, B., Cho, M.: Towards sequence-level training for visual tracking. arXiv:abs/2208.05810 (2022)
51. Fu, Z., Fu, Z., Liu, Q., Cai, W., Wang, Y.: Sparsett: visual tracking with sparse transformers. IJCAI (2022)
52. Guo, D., Shao, Y., Cui, Y., Wang, Z., Zhang, L., Shen, C.: Graph attention tracking. CVPR, pp. 9538–9547 (2020)
53. Zhao, H., Wang, D., Lu, H.: Representation learning for visual object tracking by masked appearance transfer. In: CVPR, pp. 18696–18705 (2023)
54. Zhang, Z., Liu, Y., Wang, X., Li, B., Hu, W.: Learn to match: automatic matching network design for visual tracking. ICCV, pp. 13319–13328 (2021)