Light Siamese Network for Long-Term Onboard Aerial Tracking

Xin Yang, Jinxiang Huang, Yizhao Liao, Yong Song, Ya Zhou, and Jinqi Yang

***Abstract*— The scarce onboard computational resources and real-time demand restrict the deployment of aerial trackers with sophisticated structures and customized operators. Meanwhile, aerial trackers need updating modules to adapt to continuous appearance variations in real-world long-term (LT) scenarios. However, frequent updating will introduce noisy templates and lead to tracking drifts and efficiency drops. Therefore, in this work, we develop a lightweight and highly efficient Siamese tracker for LT onboard aerial tracking applications. First, we build a compact, plain, and deployment-friendly Siamese network based on re-parameterization (Rep) as the baseline short-term (ST) tracker. Furthermore, we propose a tracking-specific decoupled knowledge distillation (KD) guided by strict teachers to unleash the appearance representation potential of the feature extractor without extra inference cost. Specifically, before distillation, the teacher conducts qualification verification to avoid misguiding the student. Then, hard negative background regions are mined and decoupled with the target region, encouraging the student to focus more on similar distractors and informative areas. Finally, to realize efficient and high-confidence LT tracking, we design two extensions and incorporate them into the boosted ST tracker: an initial-template-driven template updater with a corresponding pair-generating strategy to alleviate appearance pollution, and a confidence estimating branch to determine whether to update. Extensive results on large-scale drone benchmarks indicate that our proposed tracker significantly outperforms state-of-the-art (SOTA) aerial trackers. Real-world tests on our customized drone-captured LT dataset also validate its favorable practicability with a real-time speed of 44 fps on the Lynix KA200 chip.**

***Index Terms*— Aerial tracking, knowledge distillation (KD), model update, Siamese network.**

I. Introduction

REMOTE sensing have witnessed tremendous advancement due to the continuous expansion of multimodal data [1], [2] and the development of foundation models [2], [3], such as spectral data [4] and synthetic aperture radar data [5]. Similarly, being benefited from powerful fundamental models such as Siamese network [6], [7], [8] and vision Transformer [9], [10], [11], visual object tracking on unmanned aerial vehicle (UAV) systems has achieved significant progress. As a result, aerial tracking has possessed broad remote sensing applications, ranging from intelligent traffic surveillance [12], [13], disaster management [14] to environmental monitoring [15] and land cover mapping [1], [16]. However, lightweight model design for real-time deployment and high-confidence model update for long-term (LT) adaption is still two tricky challenges that heavily hinder the application of onboard aerial tracking.

First, the computational resources of the UAV platform are extremely scarce, which restricts the employment of the models with sophisticated structures and large scale. Referring to [17], additional branches and connections (e.g., residual addition in ResNet [18]) and special convolution (Conv) operators (e.g., depth-wise Conv in MobileNet [19]) are inefficient on hardware. Accordingly, although LightTrack [20] utilizes network structure search (NAS) and depth-wise separable Conv to decrease the amount of floating-point operations (FLOPs), the actual improvement of inference speed is limited. More importantly, some customized operators even cannot be supported on edge computation systems, such as cross-convolution and other complex correlation operators, which are the core components of the Siamese tracker. As a result, most aerial trackers [7], [9], [21] still adopt hand-craft feature extractors or early convolutional neural networks (CNNs) like AlexNet [22].

To address the real-time onboard deployment issue, we first build a compact and deployment-friendly Siamese network based on re-parameterization (Rep) as the baseline short-term (ST) tracker. As an advanced modern backbone network with both high efficiency and outstanding performance, re-parameterization VGG network (RepVGG) [17] believes that an efficient structure should be plain without any branches and only involves 3×3 convolutions, which is usually highly optimized on edge computation systems. Accordingly, it proposes a Rep residual block that can be equivalently converted to a single 3×3 convolution layer in inference. Following its structure guidelines, we not only build our tracker merely using sequential Rep blocks as far as possible but also adopt RepVGG as the backbone.

Furthermore, we propose a tracking-specific decoupled hint learning manner guided by a strict teacher to fully unleash the representation potential of the feature extractor. As discussed above, some efforts have been made to decrease the parameters of visual trackers, such as efficient Transformer [10] and NAS [20], [23]. However, another efficient way of model

Manuscript received 27 February 2024; revised 25 March 2024 and 23 April 2024; accepted 28 April 2024. Date of publication 7 May 2024; date of current version 16 May 2024. This work was supported in part by the National Natural Science Foundation of China General Program under Grant 82272130, in part by the National Natural Science Foundation of China Key Program under Grant U22A20103, and in part by the Aeronautical Science Foundation under Grant 2023Z019072001. *(Corresponding author: Yong Song.)*

The authors are with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China (e-mail: xinyang@bit.edu.cn; huangjinxiang@bit.edu.cn; liaoyizhao@bit.edu.cn; yongsong@bit.edu.cn; zhouya@bit.edu.cn; jinqiyang@bit.edu.cn).

Data is available on-line at https://github.com/YoungZnBIT/PySOT-trial.
Digital Object Identifier 10.1109/TGRS.2024.3397916

compression, that is, knowledge distillation (KD), is overlooked in the tracking field. KD can inherit the information from a large teacher model to a compact student model without any inference costs and thus has been widely used in other vision tasks such as image classification [24], [25] and object detection [26], [27], [28].

Conventional hint learning [26] is performed by directly computing the L2 distance between the teacher model's feature map and the student's, where each point on the feature maps has the same contribution to the loss. As the background pixels are far more than the foreground, the student will pay most of its attention to the easy background area instead of the target and indistinguishable distractors, limiting the advance on discriminative power. Therefore, Zhang and Ma [27] and Yang et al. [28] propose conducting distillation only on valuable and informative areas according to attention maps and decoupled masks. However, these KD methods ignore that the teacher model is not suitable for all the cases. Although overall the teacher outperforms the student, there inevitably exist a number of scenarios where the student performs better instead or the two models both underperform. Still conducting distillation in these bad cases will harm the student model instead. Thus, before distillation, it is necessary to examine whether the teacher is competent. Based on the principle that there at most exists one valid target on the search image, we leverage ground truth (GT), teacher prediction, and student prediction simultaneously to verify the teacher's qualification and decouple foreground/hard negative background regions. Additionally, we introduce attention masks into hint losses to push the student to focus on more informative pixels and channels.

Another challenge for onboard aerial trackers is the drastic appearance variation in practical LT sequences. Therefore, the model should learn to leverage historical information online for adaptation. One common way [29], [30] is to build and maintain a memory template queue that contains many intermediate frames. Obviously, the huge computation and storage cost makes it not proper for onboard tracking. Yan et al. [31] and Cui et al. [32] propose to crop one or more dynamic templates from the latest frames and directly concatenate them with the original input. However, this way accordingly weakens the resistance of noisy templates, leading to tracking drifting and even failure. Actually, frequently updating with fixed frequency will not only increase the risk of model pollution but also bring out an efficiency drop. Consequently, some metrics are proposed to select reliable templates and filter noisy templates, including intersection over union (IoU) [30], [33], center-ness [29], [34], and tracking confidence score [31], [32].

To achieve efficient and high-confidence updating, we design a dynamic template updater and a confidence estimating branch, which we incorporate into the boosted ST baseline tracker. For simplicity, we add a dynamic template branch to capture the latest status of the target. Since the initial template is the only completely reliable information during the tracking process, we consider that how to update using the dynamic template should be controlled by the initial template. Thus, we devise an updater driven by the initial template with a corresponding image-pair-generating strategy, so that the tracker can take full advantage of the initial template to resist noisy information and model pollution during updating. We intend that, when the dynamic template is inaccurate or even incorrect, model updating will not deliver negative effects; when accurate, the tracker can gain a remarkable increase in location precision. Regarding tracking confidence, the lightweight confidence branch utilizes the search feature map, correlation map, and classification map simultaneously to conduct a binary classification task, where the labels of negative and positive pairs are 0 and 1, respectively.

Our contributions are summarized as the following.

1) We build a plain and deployment-friendly Siamese tracker with both high efficiency and competitive performance based on Rep as a powerful baseline.
2) We propose a tracking-specific decoupled KD that can filter improper examples by teacher's qualification verification (TQV) and generate hard negative and attention loss masks to make the student focus more on informative regions during distillation.
3) We implement a high-confidence and efficient LT tracker by introducing a confidence estimator to determine update frequency and a dynamic updater that can adaptively update the current template and resist noisy templates according to the initial template.
4) Extensive experimental results on both public and customized datasets demonstrate that our trackers achieve both favorable performance and superior efficiency on hardware devices against state-of-the-art (SOTA) aerial trackers.

II. Related Work

A. Visual Tracking Frameworks

At present, mainstream single object tracking algorithms can be roughly divided into three branches: online learning methods represented by discriminative correlation filter (DCF) [33], [35], [36], [37], Siamese network [38], [39], [40], [41], and vision Transformer [9], [10], [31], [42].

After Henriques et al. [35] formulated the kernelized DCF tracking paradigm, numerous DCF methods [33], [36], [37] have been proposed. Benefiting from the powerful appearance representation ability of deep features, they achieve outstanding tracking performances. However, in drone images or satellite images, the appearance of tiny objects is extremely weak. For the sake of efficiency, most aerial DCF trackers [21], [43], [44] only employ hand-craft features and still adopt the tracking framework of early works such as kernelized correlation filter (KCF) [35].

The Siamese network is another prevailing tracking framework. As the first Siamese tracker, SiamFC [38] established a fully end-to-end architecture with fully offline inference. Subsequently, SiamRPN [45] and SiamRPN++ [39] further perfected the Siamese framework by introducing the region proposal network and modern backbones with padding like ResNet [18]. From then on, Siamese tracking methods have drawn a great deal of research and many extensions have emerged, including anchor-free mechanism [34], [46], correlation operator [23], [47], [48], model update

[7], [49], [50], network structure search [20], [23], and aerial Siamese tracking [7], [41], [51].

Lately, with the great success of vision Transformer, Transformer trackers have attracted increasing attention from the community and accordingly made tremendous progress in both general object tracking [10], [31], [32], [42] and aerial tracking tasks [3], [7], [9], [11]. However, compared to the Siamese network, Transformer has expensive inference and training costs, and some operators are not well supported in onboard hardware yet.

In this work, we choose the Siamese tracking framework, owing to its deployment-friendly and fully end-to-end architecture and well balance between efficiency and accuracy.

B. Model Compression

Although recently advanced backbone networks [17], [52] with powerful representative ability continuously emerge, most of the aerial trackers [7], [9], [21] still adopt AlexNet [22] and even hand-craft features. As real-world applications yearn for real-time performance, model compression is an essential topic in the deep learning field. The pruning and quantization operations can remarkably reduce the computation complexity but also lead to an accuracy decrease. Therefore, the straightforward solution is to directly design a compact model. Yan et al. [20] presents a one-shot NAS customized for Siamese tracking. The parameters of the optimal structure are only 1/20 of the Ocean's yet their performances are comparable. Blatter et al. [10] devise an Exemplar Attention Transformer layer to replace Conv blocks, improving tracking accuracy with a little additional computation expense. Referring to [17], 3×3 Conv operation and plain structure without additional branches usually have higher efficiency on edge computation systems. Thus, to build a compact tracker with competitive performance, we not only adopt RepVGG as the backbone but also use Rep blocks.

KD [24], [25] is another representative model compression method, which can transfer the knowledge of an over-parameterized teacher to a lightweight student without changing the original model. As object detection and tracking tasks have a similar framework, we inherit and improve the hint learning manner proposed in the detection distillation framework [26] for intermediate features. Specifically, the feature map produced by the student's backbone is first processed by an adaptation layer, so that it has the same channels as the teacher's. Then, the L2 loss between the two feature maps is calculated to encourage the student to learn the feature representation manner of the teacher. Zhang and Ma [27] deem that there exists a remarkable amount of imbalance between foreground and background pixels, and the student should pay more attention to valuable and informative areas. Thus, it utilizes spatial and channel attention as the loss mask. Yang et al. [28] further suggest decoupling the foreground and background areas, making the student focus on key pixels and channels.

In this work, we devise a tracking-specific distillation to fully unleash the potential of our baseline tracker. We propose a before-distillation qualification verification to avoid misguiding the student. Similar to [27] and [28], we also generate attention masks and decouple hard negative background regions and the target region to improve the distillation effect.

C. Long-Term Tracking

Referring to [53], in practical minute-level drone sequences [54], [55], disappearance (occlusion) and appearance variation are the two most significant challenges. Correspondingly, the research of LT tracking can be divided into two main directions: The first approach [56], [57], [58] achieves global searching by integrating an extra global recovery module to generate proposals and a verifier to assess disappearance and the candidates. Alternatively, it can directly perform target-specific detection on the entire input image [59], [60]. Another approach [29], [30], [49], as known as an online update, exploits intermediate frames as the reference for appearance variation adaption. As conventional Siamese trackers are fully offline during inference, it is more urgent for them to boost the temporal correspondence between the initial frame and distant search frames. Moreover, global detecting and candidate verifying will introduce huge computational costs. Thus, in this work, we mainly focus on updating.

Over the years, various ways of leveraging historical information have been continuously proposed. Some methods use previous frames to update the initial template. For instance, Ocean [49] devises an online-learned classifier to strengthen the classification map by weighted-sum, but also introduces online backpropagation and gradient descent, which are not proper for onboard deployment. To describe global temporal contexts, TCTrack [7] proposes a temporally adaptive convolution and transformer in feature extraction and similarity refinement, respectively. Joint target and background temporal propagation (JTBP) [61] utilizes not only the temporal consistencies between templates but also the temporal coherence of objects in the background by a background temporal propagation module to discriminate distractors. Some methods establish a memory queue with corresponding complex modules and mechanisms to read and maintain it. Fu et al. [29] design a space–time memory reader to aggregate template and search features. In attention in attention tracking (AiATrack) [30], the initial frame and intermediate frames interact with the search frame in the proposed LT and ST cross-attention, respectively. However, the template queue will correspondingly cause huge computation and storage costs. Thus, for simplicity, Yan et al. [31] and Cui et al. [32] directly crop several dynamic templates as additional inputs. In adaptive fusion network with dynamic template tracking (SiamMDM) [62], the correlation maps generated by dynamic templates are fused with the initial correlation map, which is more inefficient than directly fusing template features.

Although historical information is necessary to resist appearance variations, frequent updating will instead decrease efficiency and cause model pollution due to occlusion and track drifts. A common solution is introducing a scoring head with a sigmoid activation to predict the confidence score, a floating-point scalar within the range of [0, 1]. Correspondingly, only when the score exceeds the set confidence threshold, updating will be conducted. There are mainly three forms of confidence score. Accurate tracking by

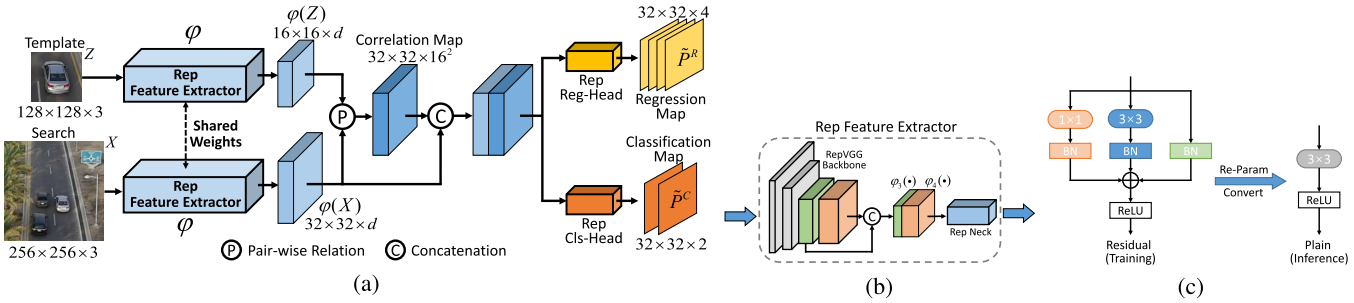


Fig. 1. Proposed lightweight baseline ST Siamese tracker based on Rep. (a) Baseline ST Siamese tracker. (b) Feature extractor. (c) Rep block.

overlap maximization (ATOM) [33] and AiATrack [30] directly predict the IoU of the bounding box, while SiamFC++ [34] and space-time memory tracking (STMTrack) [29] predict center-ness. Furthermore, spatial-temporal transformer tracking (STARK) [31] and MixFormer [32] add a simple perceptron or a learnable score token to determine whether the search frame contains a target. Accordingly, their training objective is converted to a binary classification problem, where the labels of negative and positive pairs are 0 and 1, respectively.

Considering the tradeoff between efficiency and LT adaption, we take the dynamic template scheme with a small-scale confidence head to conduct sparse and high-confidence updating. Furthermore, we think that merely using confidence scores cannot filter all noisy templates. As the initial template is the only complete information, we design an initial-template-driven updater with a corresponding training-pair-generating strategy to relieve appearance pollution.

III. Proposed Method

In this section, we build an efficient and deployment-friendly aerial tracker that can conduct adaptive and high-confidence model updating to adapt to appearance variations and resist model pollution. First, we build a compact Siamese tracker based on Rep with both high efficiency and competitive performance as the ST baseline tracker for subsequent improvement. Then, we propose a tracking-specific KD guided by a strict teacher to boost the appearance representation power of the feature extraction module without any inference cost. On the one hand, we combine the teacher output, student output, and GT to verify the qualification of the teacher and filter improper distillation examples. On the other, we generate and decouple the masks of the foreground regions, hard negative background regions, and attention maps to make the student model focus more on informative areas. Furthermore, we design a dynamic template updater with an image pair generation strategy that can adaptively update the current template and resist noisy templates according to the initial template, and a confidence estimating branch to further filter noisy templates. At last, we incorporate them into the ST tracker and implement the proposed high-confident and efficient LT tracker.

A. Compact Siamese Tracker Based on Re-Parameterization

As shown in Fig. 1, following SiamFC [38] and SiamRPN++ [39], our baseline tracker adopts the classic architecture of Siamese network. The input is an image pair: the template patch $\mathbf{Z} \in \mathbb{R}^{W_Z \times H_Z \times 3}$ which is initialized at the first frame, and the search patch $\mathbf{X} \in \mathbb{R}^{W_X \times H_X \times 3}$ cropped from the query image. First, a weight-shared backbone extracts their features. Then, the neck subnetwork refines the feature maps and reduces dimension. As the backbone and neck sequentially consist of the feature extractor part, we, respectively, denote the template and search feature map as $\varphi(\mathbf{Z}) \in \mathbb{R}^{w_Z \times h_Z \times d}$ and $\varphi(\mathbf{X}) \in \mathbb{R}^{w_X \times h_X \times d}$. Then, the two feature maps will be integrated by correlation. As shown in Fig. 2(a), conventional Siamese models usually take the cross convolution as the correlation operator, where $\varphi(\mathbf{Z})$ works as the convolution kernel and slides on $\varphi(\mathbf{X})$. Finally, the multibranch head generates the regression map $\tilde{\mathbf{P}}^R \in \mathbb{R}^{w \times h \times 4n}$, classification map $\tilde{\mathbf{P}}^C \in \mathbb{R}^{w \times h \times 2n}$, and other outputs, where n is the number of anchors or 1 (for anchor-free model). In summary, the overall inference process can be formulated as

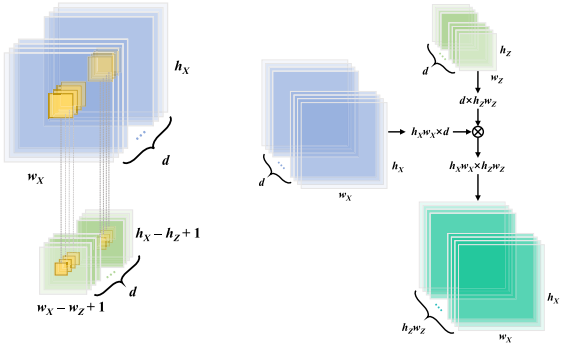
$$\tilde{\mathbf{P}}^{\tau} = \psi^{\tau}(\varphi(\mathbf{X}) \star \varphi(\mathbf{Z})) \tag{1}$$

where ψ and $\tau \in \{C, R\}$ denote the output branch and $\star$ represents the correlation operator.

To build a compact and deployment-friendly tracker, we mainly modify the following aspects of the Siamese network.

1) Rep Convolution Block: Referring to RepVGG [17], as 3×3 convolution operation is usually highly optimized on hardware, a Rep Conv block is proposed. As shown in Fig. 1(c), during training, Rep block is a residual block consisting of three parallel Conv-BN branches: a 1×1 branch, a 3×3 branch, and an identity branch; during inference, it can be completely equivalently converted to a single 3×3 Conv layer with linear rectification function (ReLU) (refer to [17] for detailed illustration and formulation). Therefore, we follow its suggestion to replace the original 3×3 Conv blocks with Rep blocks as far as possible, which can improve the model appearance representation power without any increase in parameters.

2) Backbone: We employ RepVGG [17] as the backbone, which greatly outperforms classic backbones such as AlexNet [22] and ResNet [18] with higher efficiency. More importantly, as its overall architecture is plain and only contains Rep blocks, it is more efficient and friendly to hardware. Like other modern backbones, it has multiple versions with different depths and widths. We, respectively, leverage RepVGG-A0/-A2 to build a light and a large tracker. To make it adapt to the Siamese tracking framework, we make some modifications referring to [31] and [63]. As 8 is the



(a) (b)

Fig. 2. Common correlation manners in Siamese tracking. (a) Depth-wise cross-convolution. (b) Pair-wise relation.

most proper stride for tracking, we cancel the downsampling in stage 4 and remove stage 5 to fully take advantage of multilevel features. Then, we can concatenate the outputs of stage 3 and stage 4 and send them to neck for dimension reduction. To further reduce computation cost, we in advance aggregate multilevel features in the feature extractor, rather than in head (e.g., SiamRPN++ [39]).

3) Correlation Operator: As shown in Fig. 2(b), pair-wise relation is a pixel-to-global matching manner. Referring to [23] and [47], compared with cross-convolution, pair-wise relation can obtain better performance and maintain the scale of the search feature map. Moreover, since the core operation of pair-wise relation is matrix multiplication, it is more convenient for deployment. Thus, we choose it as the correlation operator, whose computation process can be defined as

$$\mathbf{R} = \varphi(\mathbf{X}) * \varphi(\mathbf{Z})^T \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^{w_X \times h_X \times w_Z h_Z}$ is the correlation response map, $*$ denotes matrix multiplication operator, and T indicates matrix transpose. Obviously, there exists $w = w_X$ and $h = h_X$.

4) Anchor-Free Head: Following siamese box adaptive network (SiamBAN) [64] and SiamFC++ [34], we take the anchor-free mechanism, which can decrease the computational consumes of the head and avoid some sensitive hyperparameters such as anchor settings. As we adopt the template image of 128 × 128 and search image of 256 × 256, the size of $\varphi(\mathbf{Z})$ and $\varphi(\mathbf{X})$ are, respectively, 16 and 32. Thus, there are $\tilde{\mathbf{P}}^R \in \mathbb{R}^{32\times32\times4}$ and $\tilde{\mathbf{P}}^C \in \mathbb{R}^{32\times32\times2}$.

5) Inference and Training: After decoding $\tilde{\mathbf{P}}^R$ and activating $\tilde{\mathbf{P}}^C$, we can obtain the estimated bounding boxes $\tilde{\mathbf{B}} \in \mathbb{R}^{w\times h\times4}$ and positive score map $\mathbf{S} \in \mathbb{R}^{w\times h}$. Then, in the light of the GT bounding box $\mathbf{B}$, we can compute the IoU score map $\mathbf{I} \in \mathbb{R}^{w\times h}$. It is obvious that the values of $\mathbf{S}$ and $\mathbf{I}$ lie in the $[0, 1]$ interval, and the box with the highest score will be regarded as the target. In training, the overall loss $\mathcal{L} = \mathcal{L}_C + \lambda\mathcal{L}_R$ is the weighted sum of the classification loss $\mathcal{L}_C$ and regression loss $\mathcal{L}_R$.

In summary, we designed a compact Siamese tracker based on Rep with fewer parameters and better performance. Moreover, we replace cross-convolution and the anchor-based head with a pair-wise relation and an anchor-free head to further reduce parameters and deployment barriers.

B. Tracking-Specific Decoupled Knowledge Distillation Guided by Strict Teacher

KD is an effective model compression method that can improve the performance of a small model without changing the network structure. However, while KD has been widely applied in image classification and object detection, the works in the tracking field are still insufficient. In this work, we intend to employ the RepVGG-A2 tracker as the teacher to assist the RepVGG-A0 student in learning more powerful appearance representations. As the backbone occupies the majority of parameters, it is dominantly responsible for the overall tracking performance. In addition, the baseline tracker needs to be extended with additional LT tracking modules later. Thus, for convenience, we only conduct hint learning to boost the feature extractor. Notice that as a Siamese tracker generates a pair of feature maps, we only utilize the search feature map in distillation.

Conventional hint learning is conducted by directly computing the L2 distance between the feature maps of the teacher and the student. Referring to [26], an adaptation layer after the student backbone is necessary to align the student feature space with the teacher's, even though the depths are equal. Respectively, denoting the teacher and student feature map (processed by adaptation layer) as $\mathbf{F}^{\mathcal{T}} \in \mathbb{R}^{w\times h\times d}$ and $\mathbf{F}^{\mathcal{S}} \in \mathbb{R}^{w\times h\times d}$, the hint loss $\mathcal{L}_{\text{hint}}$ is

$$\mathcal{L}_{\text{hint}}\left(\mathbf{F}^{\mathcal{T}}, \mathbf{F}^{\mathcal{S}}\right) = \frac{1}{whd}\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{d}\left\|\mathbf{F}^{\mathcal{T}}_{i,j,k} - \mathbf{F}^{\mathcal{S}}_{i,j,k}\right\|^2. \tag{3}$$

Obviously, each point on the feature map has the same contribution to the loss. As the teacher cannot always perform better than the student for all of the training images, still conducting distillation when the teacher underperforms will harm the student instead. Thus, we argue that it is necessary to verify the teacher's qualifications before distillation. Furthermore, we propose performing distillation on the hard background area and foreground area, respectively, and pushing the student to focus more on informative regions.

As shown in Fig. 3, first, we generate the prediction maps of the teacher and student. We set an IoU threshold T^I and a positive score threshold T^S and then produce three Boolean masks of $w \times h$: IoU mask $\mathbf{M}^I = (\mathbf{I} > T^I)$; score mask $\mathbf{M}^S = (\mathbf{S} > T^S)$, which represents all of the positive responses; and GT mask $\mathbf{M}^G$ based on GT box $\mathbf{B}$, where only the inner points are set to 1. Obviously, when the training pair is negative, $\mathbf{M}^I$ and $\mathbf{M}^G$ will be zero matrices. As there at most exists one valid target on the search image, the inner points with high IoU and score can be viewed as the prediction of the target, while the external points with low IoU and high score are the hard negative background points where the model produces wrong responses. Therefore, we can get the foreground mask $\mathbf{M}^F = \mathbf{M}^G \cap \mathbf{M}^I \cap \mathbf{M}^S$ and hard negative background mask $\mathbf{M}^B = \overline{\mathbf{M}^G} \cap \overline{\mathbf{M}^I} \cap \mathbf{M}^S$. Then, we obtain the number of positive predictions $N_F = \sum_{i=1}^{w}\sum_{j=1}^{h}\mathbf{M}^F_{i,j}$, error points $N_B = \sum_{i=1}^{w}\sum_{j=1}^{h}\mathbf{M}^B_{i,j}$, and the average positive IoU

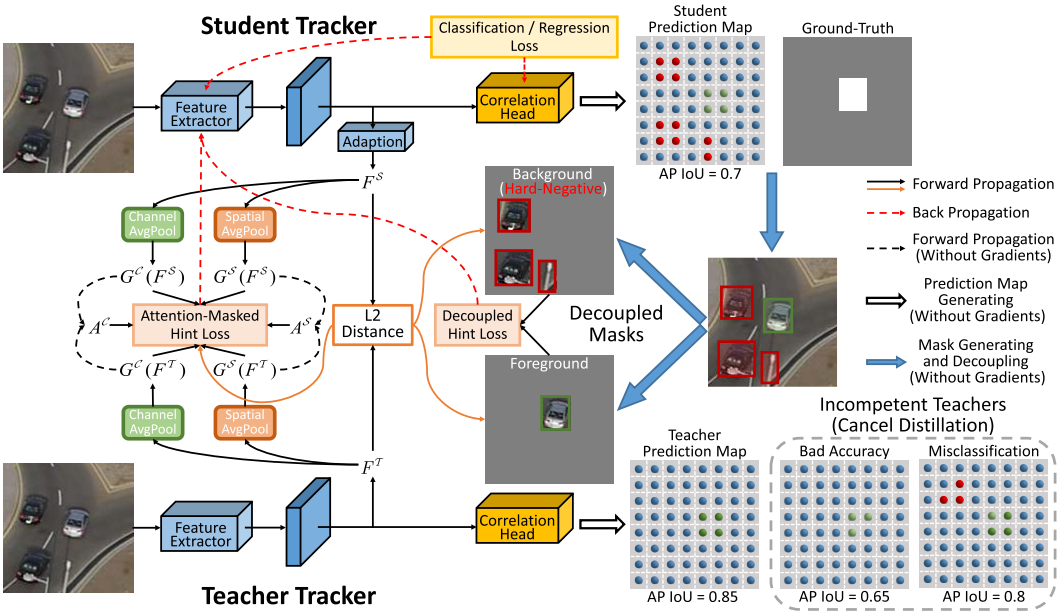


Fig. 3. Illustration of the proposed tracking-specific decoupled KD manner guided by the strict teacher.

value (AP IoU)

$$\hat{I} = \frac{1}{N_F} \sum_{i=1}^{w} \sum_{j=1}^{h} \left(\mathbf{M}_{i,j}^{F} \odot \mathbf{I}_{i,j} \right) \tag{4}$$

where $\odot$ denotes the element-wise multiplication operator.

Now, we can define the distillation condition: for the student model, it should generate wrong responses on the background region; for the teacher, its error points should be less than the error threshold T^B, and its AP IoU should exceed the student, that is, $N_B^S > 0$; $N_B^{\mathcal{T}} < T^B$; $\hat{I}^{\mathcal{T}} > \hat{I}^{\mathcal{S}}$. For instance, in Fig. 3, the student model not only generates positive responses on the target region (labeled as the green points on the prediction map) but also mistakes the similar car objects near the target and some background points (labeled as red). Meanwhile, the teacher successfully resists these distractors, and its AP IoU $\hat{I}^{\mathcal{T}} = 0.85$ surpasses the student $\hat{I}^{\mathcal{S}} = 0.7$ (the intensity of the green color represents the accuracy of predictions). However, like the instance incompetent teachers displayed in the gray dashed box (bottom right), once the teacher generates extremely inaccurate bounding boxes or redundant responses outside the target region, it will be no longer qualified for distillation (more practical failure cases are displayed in Fig. 4). Notice that all the above processes, including mask generating and decoupling, are performed without gradients.

As the prediction map gets separated into the foreground and background masks, we further decouple the hint loss so that the student can learn to distinguish the target and distractors. Furthermore, to modify the loss contribution of examples, we apply the softmax function with temperature t and mask $\mathbf{M}$ to generate normalized weight matrices

$$\text{softmax}(x_i, t, \mathbf{M}) = \frac{e^{x_i/t} \cdot M_i}{\sum_j \left(e^{x_j/t} \cdot \mathbf{M}_j \right)}. \tag{5}$$

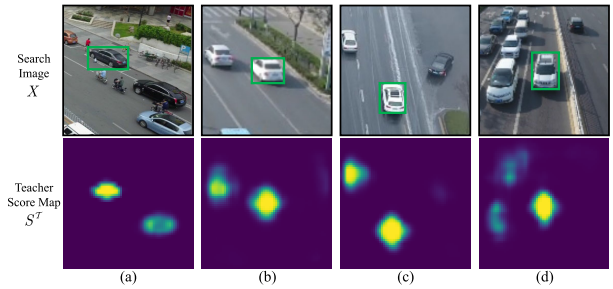


Fig. 4. Instances of incompetent teachers which produce redundant positive responses on distractors [from left to right: instance images (a)–(d)].

The foreground decoupled hint loss $\mathcal{L}_{\text{DF}}$ is computed as

$$\mathcal{L}_{\text{DF}} = \frac{1}{d} \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=1}^{d} \left(\mathbf{W}_{i,j}^{F} \cdot \left\| \mathbf{F}_{i,j,k}^{\mathcal{T}} - \mathbf{F}_{i,j,k}^{\mathcal{S}} \right\|^2 \right) \tag{6}$$

where the weight matrix $\mathbf{W}^F$ and positively correlated with the difference of the IoU values between the teacher and the student

$$\mathbf{W}^F = \text{softmax}\left(\left| \mathbf{I}^{\mathcal{T}} - \mathbf{I}^{\mathcal{S}} \right|, t^W, \mathbf{M}^{F\mathcal{S}} \right). \tag{7}$$

Similarly, we can obtain the background hint loss $\mathcal{L}_{\text{DB}}$

$$\mathcal{L}_{\text{DB}} = \frac{1}{d} \sum_{i=1}^{w} \sum_{j=1}^{h} \sum_{k=1}^{d} \left(\mathbf{W}_{i,j}^{B} \cdot \left\| \mathbf{F}_{i,j,k}^{\mathcal{T}} - \mathbf{F}_{i,j,k}^{\mathcal{S}} \right\|^2 \right) \tag{8}$$

and we hope the points with higher scores correspondingly have greater contributions to the loss

$$\mathbf{W}^B = \text{soft max}\left(\mathbf{S}^{\mathcal{S}}, t^W, \mathbf{M}^{B\mathcal{S}} \right). \tag{9}$$

Finally, the complete decoupled loss is $\mathcal{L}_{\text{DH}} = \mathcal{L}_{\text{DF}} + \mathcal{L}_{\text{DB}}$.

Additionally, as shown in Fig. 5, the attention maps generated by the pooling operation can aggregate the spatial and channel information, highlighting the regions or channels that deserve more attention. Besides, compared to the students, the teacher's attention maps are more comprehensive and detailed.

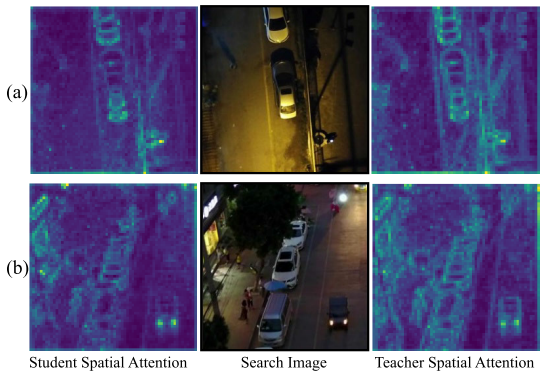


Fig. 5. Visualization of the spatial attention maps from the teacher and student tracker with corresponding images [from top to bottom: instance images (a) and (b)].

Thus, in addition to the feature map, the student should also learn to mimic the attention map. Overall, to push the student to concentrate more on informative areas, we add the attention-masked hint losses. First, calculating the spatial and channel attention maps of the teacher and student by average pooling

$$\mathbf{G}^{\mathcal{S}}(\mathbf{F}) = \frac{1}{d}\sum_{k=1}^{d}|\mathbf{F}_k| \tag{10}$$

$$\mathbf{G}^{\mathcal{C}}(\mathbf{F}) = \frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h}|\mathbf{F}_{i,j}|. \tag{11}$$

Then, generating the spatial and channel attention masks

$$\mathbf{A}^{\mathcal{S}} = d \cdot \mathrm{softmax}\left(\mathbf{G}^{\mathcal{S}}\left(\mathbf{F}^{\mathcal{T}}\right) + \mathbf{G}^{\mathcal{S}}\left(\mathbf{F}^{\mathcal{S}}\right), t^{A}\right) \tag{12}$$

$$\mathbf{A}^{\mathcal{C}} = wh \cdot \mathrm{softmax}\left(\mathbf{G}^{\mathcal{C}}\left(\mathbf{F}^{\mathcal{T}}\right) + \mathbf{G}^{\mathcal{C}}\left(\mathbf{F}^{\mathcal{S}}\right), t^{A}\right). \tag{13}$$

The attention loss $\mathcal{L}_{\mathrm{Att}}$ and attention mask loss $\mathcal{L}_{\mathrm{AM}}$ are

$$\mathcal{L}_{\mathrm{Att}} = \mathcal{L}_2\left(\mathbf{G}^{\mathcal{C}}\left(\mathbf{F}^{\mathcal{T}}\right), \mathbf{G}^{\mathcal{C}}\left(\mathbf{F}^{\mathcal{S}}\right)\right) + \mathcal{L}_2\left(\mathbf{G}^{\mathcal{S}}\left(\mathbf{F}^{\mathcal{T}}\right), \mathbf{G}^{\mathcal{S}}\left(\mathbf{F}^{\mathcal{S}}\right)\right) \tag{14}$$

$$\mathcal{L}_{\mathrm{AM}} = \sqrt{\sum_{i=1}^{w}\sum_{j=1}^{h}\sum_{k=1}^{d}\left[\mathbf{A}_k^{\mathcal{C}} \cdot \mathbf{A}_{i,j}^{\mathcal{S}} \cdot \left(\mathbf{F}_{i,j,k}^{\mathcal{T}} - \mathbf{F}_{i,j,k}^{\mathcal{S}}\right)^2\right]}. \tag{15}$$

Finally, the overall loss with distillation is formulated as

$$\mathcal{L} = \mathcal{L}_C + \lambda\mathcal{L}_R + \gamma_{\mathrm{DH}}\mathcal{L}_{\mathrm{DH}} + \gamma_{\mathrm{Att}}\mathcal{L}_{\mathrm{Att}} + \gamma_{\mathrm{AM}}\mathcal{L}_{\mathrm{AM}} \tag{16}$$

where γ_{DH}, γ_{Att}, and γ_{AM} are weights of the proposed hint losses. Note that obviously, all the hint losses will only affect the gradients on the feature extractor's parameters.

In brief, we design a tracking-specific KD guided by the strict teacher by conducting before-distillation qualification verification. Furthermore, we mine and decouple hard negative regions and re-weight their contributions to boost the discriminative power of the student trackers. In addition, we also introduce attention-based hint losses to push the student to focus more on informative areas.

C. Long-Term Tracking Extension With Dynamic Template and Confidence

As shown in Fig. 6, in this section, we design a template updater and a confidence estimating branch and incorporate them into the model to adapt to the LT tracking task.

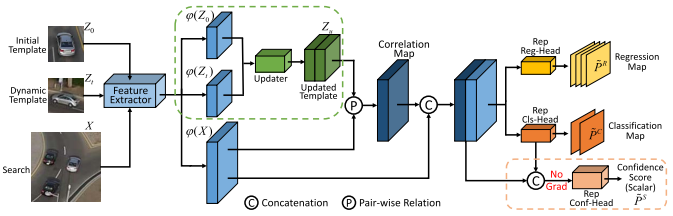


Fig. 6. Proposed LT Siamese tracker with additional template updater and confidence estimating branch.

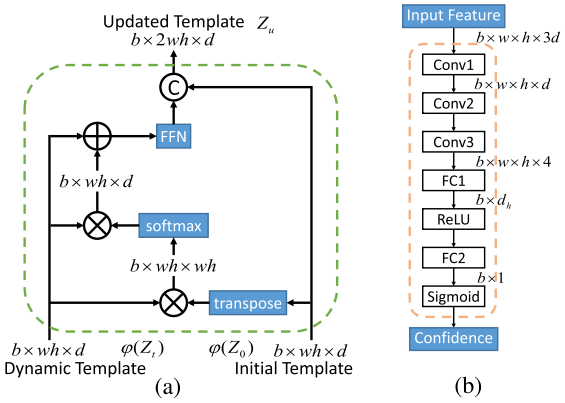


Fig. 7. Detailed structure of the proposed template updater and confidence estimating head. (a) Dynamic template updater. (b) Confidence estimating head.

To capture the latest status of the target, we adopt the dynamic template updating manner, that is, cropping a new template patch $\mathbf{Z}_t$ at the latest frame according to the estimated location and integrating it with the initial template $\mathbf{Z}_0$. However, since occlusion may occur and the location prediction may become inaccurate, frequent updating will introduce and accumulate appearance pollution, leading to tracking drift even failure. Therefore, to resist noisy templates, the usage of dynamic templates should be controlled by the initial template, which is the only completely credible information during tracking.

As shown in Fig. 7(a), in our proposed updater, we use the initial template feature map $\varphi(\mathbf{Z}_0)$ and a feed-forward network (FFN) to process the dynamic template feature $\varphi(\mathbf{Z}_t)$

$$\mathbf{Z}'_u = \mathrm{FFN}\left[\mathrm{softmax}\left[\varphi(\mathbf{Z}_t)\varphi(\mathbf{Z}_0)^T\right]\varphi(\mathbf{Z}_t) + \varphi(\mathbf{Z}_t)\right]. \tag{17}$$

The final updated template $\mathbf{Z}_u = \mathrm{Concat}(\varphi(\mathbf{Z}_0), \mathbf{Z}'_u)$ is the concatenation of the initial template $\varphi(\mathbf{Z}_0)$ and $\mathbf{Z}'_u$.

During training, dynamic templates in the input image pair are randomly sampled from adjacent frames of the search frame. In other online update trackers, when training, the dynamic templates are generated ideally like the initial template. The target lies in the center position and its size is completely accurate. However, in the piratical tracking process, the location predictions cannot always be absolutely accurate. Moreover, referring to [65], training a Siamese network needs not only positive image pairs but also negative pairs, where the targets on the template and search image are not matched. As shown in Fig. 8, we modify the generating strategy of training image pairs. First, the dynamic templates will be cropped with random translation and scale jittering to simulate the noise attributed to location error. Furthermore, to establish the priority of the initial template, pushing the

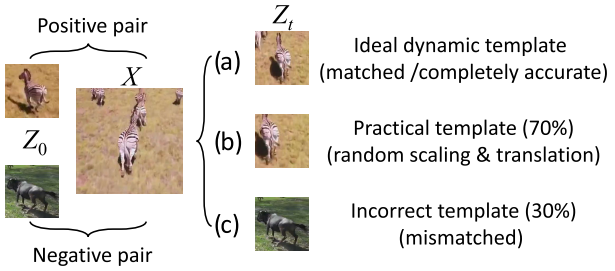


Fig. 8. Illustration of the training pair generating strategy.

model to learn to distinguish and resist noisy templates, we randomly choose 30% of the dynamic templates to get mismatched, but the property of the whole training pair is still decided by the matching condition of the initial template and search. Besides, when $\mathbf{Z}_0$, $\mathbf{X}$, and $\mathbf{Z}_t$ all match, the loss weights of these pairs will be relatively greater than other pairs in the same batch, encouraging the tracker to utilize the dynamic template to increase locating precision. In brief, we intend that proper dynamic templates can improve tracking performance while noisy templates will not introduce a negative influence.

Then, we design a confidence head to estimate the tracking confidence score of the current frame. It utilizes three clues and concatenates them into a complete input tensor: the feature map, correlation map, and classification map. Notice that to prevent the relevance from the classification head, we block the gradients of the input tensor. In other words, when training the whole LT tracker, the confidence head can be regarded as an independent module whose gradients are irrelevant to the other parts. As shown in Fig. 7(b), it consists of three sequential Rep blocks and two fully connected (FC) layers. The final output $\tilde{P}^S$ is a scalar that is compressed into a range of [0, 1] by the Sigmoid function. The training objective is a binary classification task, where the positive training pairs are labeled as 1 and the negative pairs are 0.

In the inference phase, we set an update frequency T_U and a confidence threshold T_C. Once the confidence score $\tilde{P}^S$ falls below T_C, it means that the tracking results at this frame may be no longer credible due to occlusion, out-of-view, and so on. When initializing, the initial template $\mathbf{Z}_0$ will temporarily serve as the dynamic template $\mathbf{Z}_t$, and the update interval t is set to 0. For each frame, the interval t will increase by 1. Once t exceeds T_U and meanwhile $\tilde{P}^S > T_C$, a new template $\mathbf{Z}_t$ will be cropped to update $\mathbf{Z}_u$. Then, the update interval t will be reset to 0 and restart accumulation, until the next update.

IV. Experiments

A. Implementation Details

We implement our proposed trackers and KD manner based on the Pytorch API. First, we implement two baseline ST trackers in Section III-A based on the RepVGG-A0/-A2 backbone, respectively. Then, we perform the distillation in Section III-B to boost the student backbone and obtain the final ST version of our tracker, termed MobileSiam-ST. Finally, we add the LT extensions proposed in Section III-C into MobileSiam-ST and freeze its backbone to train an LT version tracker, that is, MobileSiam-LT. Note that we take the same training settings for all above the trackers as well as the experimental trackers in ablation studies.

The IoU threshold T_I, score threshold T_S, error point threshold T_B, and confidence threshold T_C are empirically set to 0.6, 0.3, 8, and 0.45. Referring to [27], the temperatures t_W and t_A used in the softmax function are 0.4 and 0.5, and the loss weights λ, γ_{DH}, γ_{Att} and γ_{AM} are, respectively, set to 1.2, 0.02, 0.01, and 0.004. In inference, the update frequency T_U is set to 15, while in training, we randomly sample the dynamic template from a range of 25 frames around the search frame. There are 32 training pairs per mini-batch on two RTX 2080TI GPUs and 63 000 pairs per epoch. A stochastic gradient descent (SGD) optimizer with a momentum of 0.9 is employed for optimization. There are 40 epochs in total, where in the first 5 epochs the learning rate increases from 4×10^{-5} to 1×10^{-4} for a warm-up and then in the last 35 epochs it exponentially decays to 1×10^{-6}.

With regard to training datasets, to make our trackers adapt to practical aerial scenarios better, we mainly adopt drone datasets, including both a tracking dataset (VisDrone-SOT [66]) and two detection datasets (VisDrone-DET [67] and UAVDT-DET [68]). In addition, we still sample a few images from frequently used general tracking datasets as supplements, including LaSOT [69], GOT-10k [70], COCO [71], and ImageNet DET and VID [72]. All of the datasets have been split into training and test subsets by their authors before releasing.

For evaluation, we test our proposed trackers on four challenging aerial benchmarks: UAV123 [73] contains 123 videos of 30 fps which are captured from low-altitude UAVs. For better evaluation, we choose its downsampled version (10 fps), that is, UAV123@10fps. DTB70 [74] is a drone dataset that owns 70 high-diversity sequences and mainly focuses on the challenge of severe camera motion. VisDrone-SOT [66] includes 25 LT videos and ten ST videos. UAVDT-SOT [68] is the SOT subset of UAVDT and consists of 50 sequences sampled from 10-h videos. To demonstrate the advancement of our trackers more comprehensively, we further evaluate them on two recently released large-scale benchmarks: UAVTrack112 [6] contains 112 real-world videos to measure the robustness of aerial trackers under typical challenges such as fast motion and LT tracking. VTUAV [54] is a visible-thermal high-resolution drone dataset with diverse categories and scenes. All the benchmarks follow the one-pass evaluation protocol and measure trackers in terms of the area under the curve (AUC) of the success plot and precision score.

B. Ablation Study

1) Version Analysis: In this work, we implement four versions of the tracker: a baseline tracker with the RepVGG-A0 backbone as the **student** in KD (**MobileSiam-A0**); a tracker with RepVGG-A2 as the **teacher** (**MobileSiam-A2**); the boosted ST tracker (**MobileSiam-ST**); the LT tracker with LT extensions (**MobileSiam-LT**). In Table I, we report their computation complexity (including parameter number, GFLOPs, and average speed on RTX2080Ti GPU) and tracking performance (AUC score on UAV123@10fps and DTB70), which are represented as the experimental trackers (3), (4), (5), and

TABLE I

COMPUTATION COMPLEXITY AND TRACKING PERFORMANCE ANALYSIS OF THE DIFFERENT VERSIONS OF OUR TRACKERS

#	Backbone	KD	LT	Params	GFLOPs	Speed	UAV10fps	DTB70
1	Mobile-v3-L			8.99	9.21	75	0.621	0.630
2	ResNet-18			22.63	21.52	89	0.611	0.652
3	RepVGG-A2			20.86	21.81	109	0.672	0.698
4	RepVGG-A0			11.76	12.19	147	0.592	0.607
5	RepVGG-A0	✓		11.76	12.19	147	0.614	0.631
6	RepVGG-A0	✓	✓	17.87	15.77	139	0.635	0.654

(6). In addition, to validate the advancement of the RepVGG backbone, we also implement two baseline ST trackers using the MobileNet v3 Large [19] (Mobile-v3-L) and ResNet-18 [18] backbone as the trackers (1) and (2).

Although MobileNet tracker (1) seems to be the lightest version and meanwhile performs well, counterintuitively, its actual speed is the last place, merely half of MobileSiam-ST. This may be attributed to the depth-wise separable convolution operation, which is parameter-economical yet inefficient for edge computation systems. As the most heave model, the speed and performance of the ResNet-18 tracker (2) are all far inferior to MobileSiam-A2 (3). Actually, a huge performance gap exists between all trackers and tracker (3), which indicates that it can transfer a substantial amount of knowledge to the student model during distillation. As the MobileSiam-A0 tracker owns an excellent balance between speed and tracking performance, we take it as the baseline tracker and unleash its representation potential by distillation and introducing LT extensions. A more detailed analysis of the effect of KD and LT will be discussed in subsequent experiments. Note that although the computational consumption of the LT tracker seems to increase greatly, as it only estimates tracking confidence and conducts updating every T_U frames, the practical decline in speed is slight (referring to Table II).

2) Tracking-Specific Knowledge Distillation: In this section, we conduct an ablation study to analyze the impact of each component in the decoupled KD manner guided by the strict teacher. The proposed KD manner can be divided into three components: TQV, the decoupled hint loss $\mathcal{L}_{\text{DH}}$ and the attention-masked hint losses $\mathcal{L}_{\text{Att}}$ and $\mathcal{L}_{\text{AM}}$. Table III reports the AUC scores of the experimental trackers on UAV123@10fps, VisDrone-SOT, and DTB70.

The results of "dist-abl1" and "dist-abl3" indicate that even if only applying $\mathcal{L}_{\text{DH}}$ or $\mathcal{L}_{\text{Att}}+\mathcal{L}_{\text{AM}}$, all three types of hint loss can slightly enhance the appearance representation power of the feature extractor. Then, "dist-abl2" and "dist-abl4" show that if verifying the teacher before distillation, the model can gain a greater performance increase. Finally, our proposed light Siamese tracker's ST version, which is boosted by the complete KD manner, surpasses the baseline by 0.022, 0.024, and 0.016, respectively.

In addition, we also display several classification response maps of the student tracker to visualize the effectiveness of the proposed decoupled distillation. As shown in Fig. 9, before distillation, the student always produces redundant responses on distractors with a similar appearance. Fortunately, in our proposed KD, these distractors will be mined and decoupled to boost the discriminative power. Consequently, after distillation,

TABLE II

INCREMENTAL COMPONENTWISE ANALYSIS OF THE PROPOSED LT TRACKING EXTENSIONS ON UAV123@10fps AND UAVDT USING THE AUC METRIC

Components	UAV123@10fps	UAVDT	Speed
ours-ST	0.614	0.588	147.13
(+ Directly Concatenate)	0.608 (-0.006)	0.590 (+0.002)	142.21
+ Template Updater	0.621 (+0.007)	0.608 (+0.020)	140.61
+ Training Pair	0.626 (+0.005)	0.615 (+0.007)	140.61
ours-LT (+ Confidence)	0.635 (+0.009)	0.627 (+0.012)	139.38

TABLE III

ABLATION STUDY OF THE PROPOSED DECOUPLED KD MANNER. ALL EXPERIMENTAL TRACKERS ARE EVALUATED ON UAV123@10fps, VISDRONE-SOT, AND DTB70 IN AUC

	TQV	$\mathcal{L}_{DH}$	$\mathcal{L}_{Att}+\mathcal{L}_{AM}$	UAV10fps	DTB70	VisDrone
baseline	-	-	-	0.592	0.607	0.595
dist-abl1			✓	0.599	0.611	0.597
dist-abl2	✓		✓	0.607	0.619	0.608
dist-abl3		✓		0.603	0.615	0.599
dist-abl4	✓	✓		0.610	0.624	0.604
ours-ST	✓	✓	✓	**0.614**	**0.631**	**0.611**

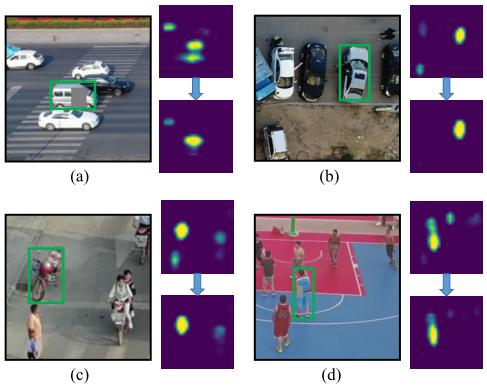


Fig. 9. Response maps from the student tracker before and after distillation, i.e., baseline versus MobileSiam-ST [from left to right: instance images (a)–(d)].

the responses of MobileSiam-ST are much cleaner. It is also worth mentioning that as the student performance grows, the ratio of no-distillation examples correspondingly increases, owing to the cases where the student accuracy is comparable and even better than the teacher's. For instance, in epoch 5, there are 12.51% of examples whose distillation losses are set to 0 (4.20% due to misclassification and 8.31% due to inaccuracy), while in epoch 30, the ratio rises to 16.75% (4.24% for misclassification and 12.51% for inaccuracy). Thus, the dynamic change and increase in the abandoned example ratio align with the distillation condition of the proposed TQV in Section III-B.

3) Long-Term Extensions: As shown in Table II, we perform an incremental componentwise experiment on UAV123@10fps and UAVDT in terms of AUC and fps to analyze the effectiveness and time cost of these LT extensions proposed in Section III-C.

Here, the ST version of our tracker, whose backbone is boosted by the proposed KD manner, serves as the baseline tracker. Moreover, to fully demonstrate the advantage of the template updater, we add an experimental tracker in the second row, which directly concatenates the initial

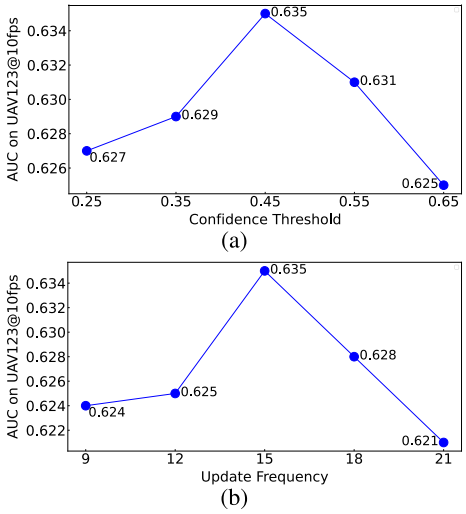


Fig. 10. Effect of the confidence threshold T_C and update frequency T_U using the AUC score on UAV123@10fps. (a) Performance varies when T_S changes in $[0.25, 0.65]$. The interval is 0.05 and $T_u = 15$. (b) Performance varies when T_I changes in $[9, 21]$. The interval is 3 and $T_c = 0.45$.

and dynamic template as the updated template without any processing, similar to the updated style of Stark [31]. The update frequency T_U is set to 15 for all the experimental trackers. The comparison of the first three rows proves the necessity for model updating, and the template updater is more effective than directly concatenating. Then, by adding the corresponding image-pair-generating strategy in training, the tracker can achieve an increase of 0.005/0.007 without any decline in speed. After equipping the confidence branch, the final LT tracker outperforms the ST baseline by 0.021/0.039 on the two aerial datasets with a slight speed decrease of 9 fps, emphasizing the importance of the update check.

4) Effect of Long-Term Tracking Thresholds: There are two key thresholds that are manually set to control model updating: the confidence threshold T_C and update frequency T_U. Intuitively, as the values of T_C and T_U are higher, the update criteria become stricter, resulting in sparser updates. Although too frequent updates can increase the risk of model pollution and computational complexity, the tracker also needs to adapt promptly to the target appearance variations. Therefore, we conduct experiments to analyze the effect of these two LT tracking thresholds.

MobileSiam-LT is adopted as the experimental tracker and evaluations are performed on UAV123@10fps using the AUC metric. We design the following two experiments to observe the effect of the two thresholds.

1) T_C takes values from a range of $[0.25, 0.65]$ with an interval of 0.10, while T_U remains at 15.
2) T_U takes values from a range of $[9, 21]$ with an interval of 3, while T_C remains at 0.45.

As Fig. 10 shows, the performance variation trends of T_C and T_U are all consistent with the above analysis. Namely, both excessively sparse and overly frequent updates can lead to a decrease in tracking performance. Specifically, compared to the performance at a confidence threshold of 0.45, the AUC score slightly decreased by 0.10/0.08 when T_C increased to 0.65 or decreased to 0.25. A similar performance curve was observed for the update frequency T_U.

TABLE IV

PERFORMANCE AND SPEED COMPARISON OF OUR TRACKERS AND SEVEN SOTA TRACKERS ON DTB70. ALL THE SPEEDS ARE EVALUATED ON A SINGLE 2080TI GPU WITH A BATH SIZE OF 1

Tracker	AUC ↑	Pr ↑	Speed ↑
Ocean-online	0.455	0.636	34.9
SiamFC++	0.637	0.814	90.0
AutoTrack	0.477	0.718	65.4
HiFT	0.594	0.804	129.8
TCTrack++	0.626	0.815	99.2
SiamAPN++	0.594	0.729	128.7
SmallTrack	**0.654**	**0.858**	72.5
MobileSiam-ST	0.631	0.826	**147.1**
MobileSiam-LT	**0.654**	0.847	139.4

Briefly, our tracker shows low sensitivity to LT tracking hyperparameters, and we choose the optimal parameter combination of $T_C = 0.45$ and $T_U = 15$.

C. Comparison With the State-of-the-Art

We compare the ST and LT versions of our proposed tracker with 12 representative SOTA trackers, including one Transformer general tracker exemplar transformers tracking (ETTrack) [10], two Transformer aerial tracker hierarchical feature transformer (HiFT) [9] and saliency-guided dynamic vision transformer (SGDViT) [11], one correlation-filter-based aerial tracker AutoTrack [75], four aerial Siamese trackers SiamAPN++ [41], TCTrack++ [76], SmallTrack [8], and JTBP [61], and six general Siamese trackers: SiamRPN++ [39], SiamFC++ [34], SiamGAT [48], Ocean [49], UpdateNet [50], and LightTrack [20]. Aerial trackers (and UpdateNet) prefer lightweight and deployment-friendly backbones like AlexNet [22]. LightTrack further leverages network architecture search to build a customized tiny-scale backbone. General Siamese trackers usually employ deeper backbone such as GoogLeNet [77] (SiamFC++ and SiamGAT) and ResNet50 [18] (SiamRPN++ and Ocean). In addition, AutoTrack, TCTrack++, Ocean, and UpdateNet also have an online update module for LT tracking.

1) Overall Performance: As shown in Fig. 11, our proposed MobileSiam-ST and MobileSiam-LT both perform favorably on all four benchmarks against the above SOTA trackers. Moreover, we also evaluate the speed of some comparison trackers on a single 2080TI GPU with a bath size of 1 and report their speeds in Table IV.

As the appearance representation power of the feature extractor gets boosted by the strict-teacher-guided decoupled KD, MobileSiam-ST achieves competitive performance against lightweight trackers and other aerial trackers such as LightTrack and SiamAPN++. On DTB70 and UAVDT, MobileSiam-ST is slightly inferior to some recent aerial trackers like JTBP and general trackers with a deeper backbone like SiamFC++. After being equipped with the proposed LT extensions, MobileSiam-LT comprehensively outperforms these comparison trackers in both AUC and precision on all benchmarks, especially online update trackers like Ocean and TCTrack++.

Specifically, on UAV123@10fps, DTB70, and UAVDT, MobileSiam-LT realizes the AUC scores of 0.635/0.654/0.626,

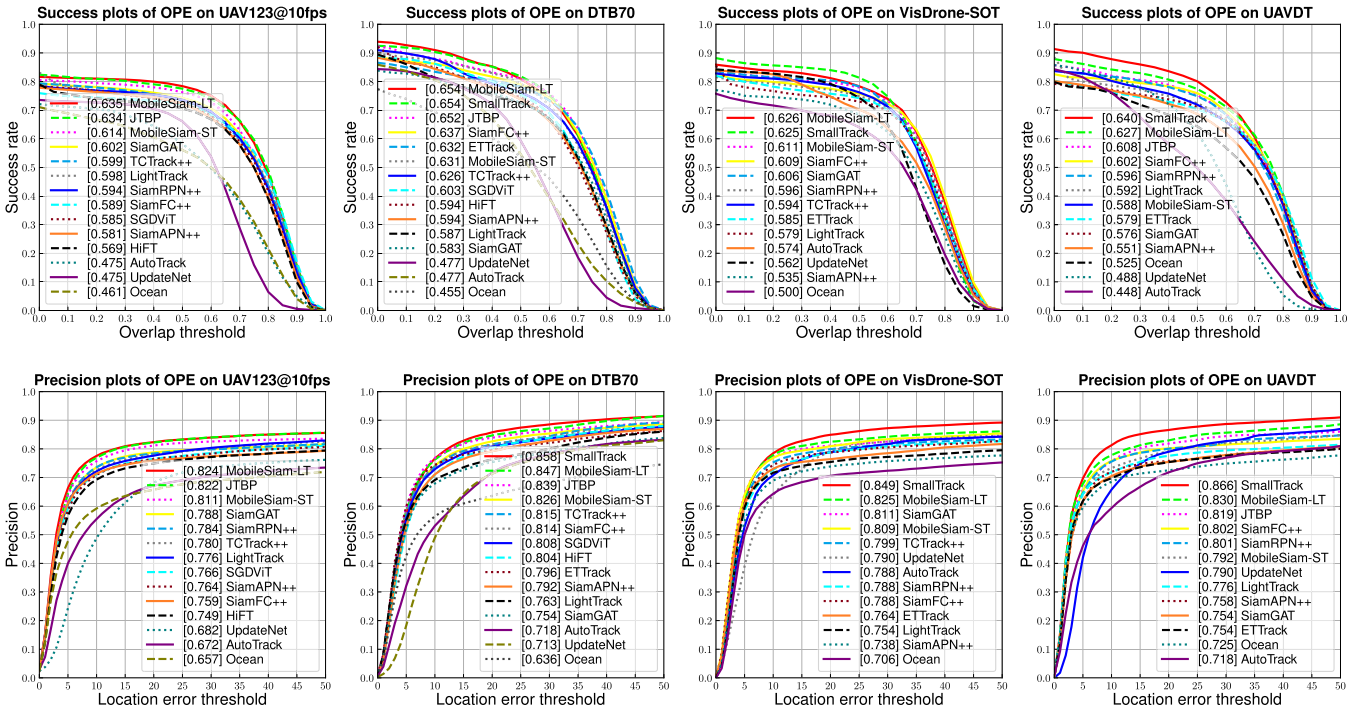


Fig. 11. Overall performance comparison using the AUC and precision metrics on four challenging large-scale aerial tracking benchmarks: UAV123@10fps, DTB70, VisDrone-SOT, and UAVDT. MobileSiam-ST and MobileSiam-LT, respectively, denote the short- and long-term versions of our proposed tracker.

respectively, and takes the first place. Although SmallTrack surpasses MobileSiam-LT slightly by 0.013 in terms of AUC, as shown in Table IV, our trackers run at twice the speed of SmallTrack and are faster than all other trackers.

2) Attribute-Based Evaluation: As shown in Fig. 12, we display the tracking performance on four attributes on UAV123@10fps: illumination variation, viewpoint change, scale variation, and aspect ratio change. These four attributes mainly represent the appearance variation challenges in LT scenarios caused by background, camera position, or the target itself. MobileSiam-LT still ranks first place on all attributes and remarkably improves the performance of MobileSiam-ST by 2.5%, 8.6%, 3.7%, and 6.8%, which highlights the effectiveness of our proposed LT modules again.

3) Evaluation on Recent Benchmarks: To demonstrate the advancement of our trackers more comprehensively, we further evaluate them on two recently released large-scale benchmarks: UAVTrack112 [6] and VTUAV [54]. For a fair comparison, we only test our trackers on the RBG LT test subset sequences of VTUAV against other RBG-only trackers. As shown in Fig. 13, our trackers still outperform SOTA trackers on UAVTrack112. For instance, our MobileSiam-LT surpasses SiamFC++ by a gap of 0.019/0.035 in terms of AUC and precision. Furthermore, as 50 videos of UAVTrack112 are LT, our LT tracker improves the performance of the ST tracker by 0.027 and 0.039, respectively, which indicates the effectiveness of our proposed LT tracking modules again.

Regarding the LT benchmark VTUAV, as our trackers lack modules for occlusion detection and global searching, their performances are slightly inferior to the SPLT tracker which is specialized for disappearance cases. However, our MobileSiam-LT is still competitive against other trackers

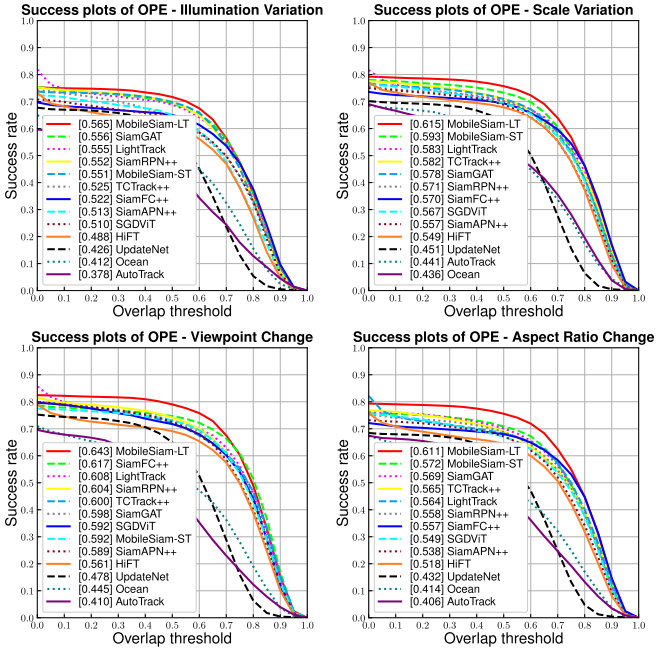


Fig. 12. AUC scores of SOTA trackers on four attributes related to appearance variation on UAV123@10fps.

without recovery modules such as HiFT and Ocean. Thus, we believe that our MobileSiam-ST can be a strong candidate for the local tracker in LT tracking frameworks such as [56] and [57] owing to its simple structure and high efficiency

D. Real-World Tests

To validate the performance and efficiency of the proposed light Siamese tracker in real-world LT UAV-captured scenarios, we deploy MobileSiam-LT on the Lynxi KA200[1]

[1]https://www.lynxi.com/lingqiKA200/18.html

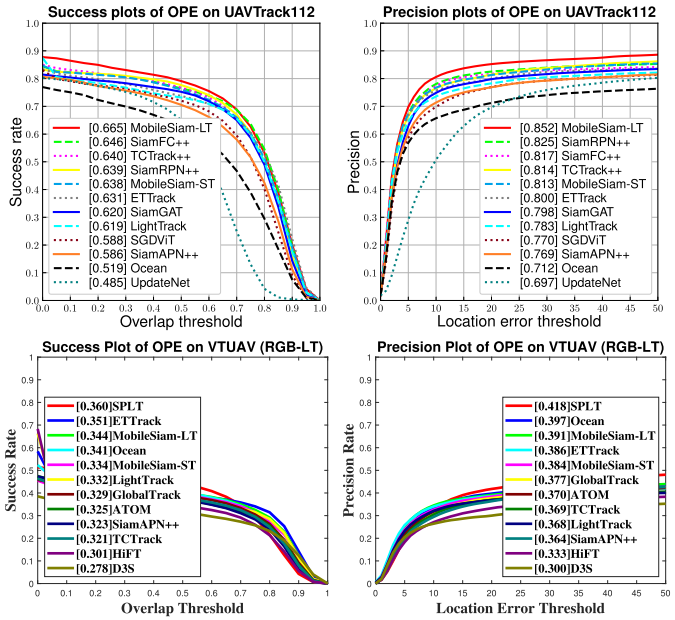


Fig. 13. Overall performance comparison using the AUC and precision metrics on two recent drone benchmarks: UAVTrack112 and VTUAV (only on RGB LT videos).

brain-like chip and test it on our customized aerial dataset. There are a total of seven sequences where targets are different kinds of vehicles. The sequences are captured in the wild by a Zenmuse H20T zoom camera mounted on our DJI-M300-RTK[2] UAV in a resolution of 1920 × 1080, flying at altitudes between 50 and 120 m (the maximum allowed altitude). Besides, we compare our teacher model MobileSiam-A2 and MobileSiam-LT against two SOTA aerial trackers: SiamAPN++ [41] and TCTrack++ [39], and a representative Siamese tracker with online updating: Ocean [49], all of which are still performed on GPU. In addition to AUC score and precision, we also report their average center location error (CLE) and utilize fps to measure efficiency.

As shown in Table V, our MobileSiam-LT remarkably outperforms all the SOTA trackers on all accuracy metrics, especially for CLE, which means that it hardly ever loses the target in these practical drone-captured videos. Meanwhile, it also runs at a high speed of 44 fps on the KA200 chip, meeting the real-time deployment condition. Moreover, it is noteworthy that although the teacher tracker MobileSiam-A2 greatly outperforms MobileSiam-LT on public aerial benchmarks such as UAV123@10fps, it is inferior in real-world LT scenarios, proving the importance of online updating.

Moreover, as shown in Fig. 14, we select and exhibit five qualitative comparison results to illustrate why our tracker performs better. From the top to the bottom, the test videos are labeled as (a) to (e). To demonstrate the challenges in these videos more clearly through merely three frames, we do not show the initial frame. In these real-world LT sequences, during the target motion process, as the angle and distance between the drone and target change, the target scale undergoes a dramatic change process of first increasing and then decreasing, and accordingly some detailed target appearance information will continuously appear and disappear. For

TABLE V

PERFORMANCE COMPARISON OF SIAMAPN++, TCTRACK++, OCEAN-ONLINE, AND OUR MOBILESIAM-A2 AND MOBILESIAM-LT ON OUR CUSTOMIZED WILD AERIAL DATASET IN TERMS OF CLE, AUC SCORE, PRECISION (PR), AND FPS

Tracker	Deployment Platform	CLE ↓	AUC ↑	Pr ↑	Speed ↑
Ocean-online	RTX 2080Ti GPU	171.66	0.679	0.774	40.61
SiamAPN++	RTX 2080Ti GPU	100.48	0.577	0.681	128.71
TCTrack++	RTX 2080Ti GPU	124.46	0.626	0.720	99.24
MobileSiam-A2	KA200 Brain-like Chip	23.94	0.797	0.944	23.15
MobileSiam-LT	KA200 Brain-like Chip	**6.95**	**0.811**	**0.944**	43.73

Fig. 14. Visualization of our MobileSiam-LT, MobileSiam-A2, SiamAPN++, TCTrack++, and Ocean on five LT testing sequences of our customized real-world drone-captured dataset. Better viewed with zoom-in in color.

instance, in videos (b) and (e), once the scale and angle of the target bus and truck change, the ST tracker SiamAPN++ fails to adapt and its predicted bounding boxes are more inaccurate than other trackers.

In addition, the background always contains numerous occlusions such as plants and distractors with similar appearances. Thus, when the target appearance changes significantly, despite equipping model update modules, the comparison trackers, such as Ocean and SiamAPN++, are easy to confuse the target with distractors. Benefiting from the initial-template-driven high-confidence update, our MobileSiam-LT can adapt to the aforementioned drastic scale variation and partial occlusion. Moreover, the decoupled distillation manner guided by a strict teacher also helps resist similar distractors.

However, as there always exist delays in camera motion, the targets partially or even completely move out of view frequently, especially in videos (a) and (d). Although our tracker recovers the target coincidentally as the positions where the target disappears and reappears are close, it indeed cannot deal with more challenging cases due to the lack of global search modules. Since TCTrack++ only introduces adaptive temporal modules to capture the temporal context but does not estimate tracking confidence, it is polluted by background during disappearance. Thus, it fails to recover the target after it reappears. Additionally, despite MobileSiam-LT

[2]https://www.dji.com/matrice-300

showing powerful adaptability for scale variation, it is also influenced by the white van like other trackers in video (e).

Briefly, the quantitative and qualitative experimental results indicate that our proposed tracker owns powerful adaptability for appearance variations and discrimination for distractors on practical LT aerial scenarios with high speed.

V. Conclusion

In this work, we are devoted to developing a lightweight and high-confidence Siamese tracker for LT aerial tracking applications, which can run at high speed and be friendly for onboard deployment. First, we build a compact and plain Siamese network based on Rep as the baseline ST tracker. Then, we propose a tracking-specific decoupled KD manner guided by a strict teacher to enhance the appearance representation and discriminative power of the feature extractor without any inference cost. Finally, we devise an initial-template-driven updater with a corresponding training-image-pair-generating strategy and confidence estimating branch and incorporate them into the boosted ST tracker to adapt to the target appearance variations in LT sequences. Extensive results on both large-scale drone benchmarks and our customized dataset indicate that our tracker can favorably handle the real-world LT aerial onboard tracking task in remote sensing applications with high efficiency. Therefore, we are convinced that our tracker can inspire further research to deal with the LT tacking problem and the proposed KD manner is also effective for other aerial trackers. However, limited by computation resources, we have not designed modules for occlusion detection and recovery and the model updating manner is also simple. In the future, we will continue to explore building a complete LT tracking framework while keeping the efficiency of the model.

References

[1] D. Hong et al., "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, Aug. 2020.

[2] X. He, Y. Chen, L. Huang, D. Hong, and Q. Du, "Foundation model-based multimodal remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5502117.

[3] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 3, 2024, doi: 10.1109/TPAMI.2024.3362475.

[4] D. Hong, J. Yao, C. Li, D. Meng, N. Yokoya, and J. Chanussot, "Decoupled-and-coupled networks: Self-supervised hyperspectral image super-resolution with subpixel fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5527812.

[5] F. Ma, X. Sun, F. Zhang, Y. Zhou, and H.-C. Li, "What catch your attention in SAR images: Saliency detection based on soft-superpixel lacunarity cue," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5200817.

[6] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Onboard real-time aerial tracking with efficient Siamese anchor proposal network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5606913.

[7] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, and C. Fu, "TCTrack: Temporal contexts for aerial tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14798–14808.

[8] Y. Xue et al., "SmallTrack: Wavelet pooling and graph enhanced classification for UAV small object tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5618815.

[9] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "HiFT: Hierarchical feature transformer for aerial tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 15457–15466.

[10] P. Blatter, M. Kanakis, M. Danelljan, and L. V. Gool, "Efficient visual tracking with exemplar transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 1571–1581.

[11] L. Yao, C. Fu, S. Li, G. Zheng, and J. Ye, "SGDViT: Saliency-guided dynamic vision transformer for UAV tracking," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 3353–3359.

[12] C. Fu, B. Li, F. Ding, F. Lin, and G. Lu, "Correlation filters for unmanned aerial vehicle-based aerial tracking: A review and experimental evaluation," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 125–160, Mar. 2022.

[13] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 91–124, Mar. 2022.

[14] C. Yuan, Z. Liu, and Y. Zhang, "Aerial images-based forest fire detection for firefighting using optical remote sensing techniques and unmanned aerial vehicles," *J. Intell. Robotic Syst.*, vol. 88, nos. 2–4, pp. 635–654, Dec. 2017.

[15] X. Qiao, Y. Zhao, L. Chen, S. G. Kong, and J. C.-W. Chan, "Mosaic gradient histogram for object tracking in DoFP infrared polarization imaging," *ISPRS J. Photogramm. Remote Sens.*, vol. 194, pp. 108–118, Dec. 2022.

[16] A. Jamali, S. K. Roy, D. Hong, P. M. Atkinson, and P. Ghamisi, "Spatial-gated multilayer perceptron for land use and land cover mapping," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.

[17] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13733–13742.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[20] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, "LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15180–15189.

[21] C. Fu, J. Ye, J. Xu, Y. He, and F. Lin, "Disruptor-aware interval-based response inconsistency for correlation filters in real-time aerial tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6301–6313, Aug. 2021.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[23] Z. Zhang, Y. Liu, X. Wang, B. Li, and W. Hu, "Learn to match: Automatic matching network design for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13339–13348.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[25] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11962.

[26] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[27] L. Zhang and K. Ma, "Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.

[28] Z. Yang et al., "Focal and global knowledge distillation for detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4643–4652.

[29] Z. Fu, Q. Liu, Z. Fu, and Y. Wang, "STMTrack: Template-free visual tracking with space-time memory networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13774–13783.

[30] S. Gao, C. Zhou, C. Ma, X. Wang, and J. Yuan, "AiATrack: Attention in attention for transformer visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 146–164.

[31] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10448–10457.

[32] Y. Cui, C. Jiang, L. Wang, and G. Wu, "MixFormer: End-to-end tracking with iterative mixed attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13608–13618.

[33] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669.

[34] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 12549–12556.

[35] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Feb. 2014.

[36] A. Lukežic, J. Matas, and M. Kristan, "D3S—A discriminative single shot segmentation tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7131–7140.

[37] C. Mayer, M. Danelljan, D. P. Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13444–13454.

[38] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. ECCV Workshops*, Oct. 2016, pp. 850–865.

[39] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

[40] X. Yang, C. Zhao, J. Yang, Y. Song, and Y. Zhao, "Negative-driven training pipeline for Siamese visual tracking," *IEEE Trans. Multimedia*, vol. 26, pp. 4416–4429, 2024.

[41] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "SiamAPN++: Siamese attentional aggregation network for real-time UAV tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 3086–3092.

[42] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 8126–8135.

[43] W. Zhang, L. Jiao, F. Liu, L. Li, X. Liu, and J. Liu, "MBLT: Learning motion and background for vehicle tracking in satellite videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4703315.

[44] S. Xuan et al., "Rotation adaptive correlation filter for moving object tracking in satellite videos," *Neurocomputing*, vol. 438, pp. 94–106, May 2021.

[45] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[46] X. Yang, Y. Song, Y. Zhao, Z. Zhang, and C. Zhao, "Unveil the potential of Siamese framework for visual tracking," *Neurocomputing*, vol. 513, pp. 204–214, Nov. 2022.

[47] B. Liao, C. Wang, Y. Wang, Y. Wang, and J. Yin, "PG-Net: Pixel to global matching network for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12367. Cham, Switzerland: Springer, 2020, pp. 429–444.

[48] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, and C. Shen, "Graph attention tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9543–9552.

[49] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. ECCV*, Aug. 2020, pp. 771–787.

[50] L. Zhang, A. Gonzalez-Garcia, J. Van De Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for Siamese trackers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4010–4019.

[51] W. Song et al., "A joint Siamese attention-aware network for vehicle object tracking in satellite videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5625617.

[52] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[53] C. Liu, X.-F. Chen, C.-J. Bo, and D. Wang, "Long-term visual tracking: Review and experimental comparison," *Mach. Intell. Res.*, vol. 19, no. 6, pp. 512–530, Dec. 2022.

[54] P. Zhang, J. Zhao, D. Wang, H. Lu, and X. Ruan, "Visible-thermal UAV tracking: A large-scale benchmark and new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8876–8885.

[55] C. Zhang et al., "WebUAV-3M: A benchmark for unveiling the power of million-scale deep UAV tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 9186–9205, Jul. 2023.

[56] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'Skimming-perusal' tracking: A framework for real-time and robust long-term tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2385–2393.

[57] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, "High-performance long-term tracking with meta-updater," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6298–6307.

[58] Z. Zhou, J. Chen, W. Pei, K. Mao, H. Wang, and Z. He, "Global tracking via ensemble of local trackers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8751–8760.

[59] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6578–6588.

[60] L. Huang, X. Zhao, and K. Huang, "GlobalTrack: A simple and strong baseline for long-term tracking," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11037–11044.

[61] X. Lei, W. Cheng, C. Xu, and W. Yang, "Joint target and background temporal propagation for aerial tracking," *ISPRS J. Photogramm. Remote Sens.*, vol. 211, pp. 121–134, May 2024.

[62] J. Yang, Z. Pan, Z. Wang, B. Lei, and Y. Hu, "SiamMDM: An adaptive fusion network with dynamic template for real-time satellite video single object tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5608619.

[63] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[64] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6668–6677.

[65] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 101–117.

[66] D. Du et al., "VisDrone-SOT2019: The vision meets drone single object tracking challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 199–212.

[67] D. Du et al., "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 213–226.

[68] D. Du et al., "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 370–386.

[69] H. Fan et al., "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5374–5383.

[70] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[71] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[72] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[73] M. Mueller, N. G. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 445–461.

[74] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1–7.

[75] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11923–11932.

[76] Z. Cao, Z. Huang, L. Pan, S. Zhang, Z. Liu, and C. Fu, "Towards real-world visual tracking with temporal contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15834–15849, Dec. 2023.

[77] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2015, pp. 1–9.

Xin Yang received the B.S. degree from the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree.

His research interests include deep object detection and tracking.

Jinxiang Huang received the B.S. degree from the School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, China, in 2022. He is currently pursuing the master's degree with the School of Optics and Photonics, Beijing Institute of Technology, Beijing.

His research interests include UAV object detection and knowledge distillation.

Yizhao Liao received the B.S. degree from the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree.

His research interests include brain-inspired computing hardware and high-speed object detection based on spiking neural networks.

Yong Song received the Ph.D. degree from the Department of Optoelectronic, Beijing Institute of Technology, Beijing, China, in 2004.

He is currently a Professor and the Director of the School of Optics and Photonics, Institute of Optoelectronic Instrument, Beijing Institute of Technology. He has been engaging in research on brain-inspired intelligence and intelligent interaction.

Dr. Song is an Executive Director of the Image Science and Engineering Chapter, China Instrument and Control Society (CIS), and the Editorial Board Member of the *Journal of Optical Technology* and *Journal of Ordnance Equipment Engineering*.

Ya Zhou received the Ph.D. degree from the Department of Optoelectronic, Beijing Institute of Technology, Beijing, China, in 2000.

She is currently an Associate Professor with the School of Optics and Photonics, Institute of Optoelectronic Instrument, Beijing Institute of Technology. Her research interests include biometric and visual measurement.

Jinqi Yang received the B.E. degree from Beijing Institute of Technology, Beijing, China, in 2021, where she is currently pursuing the M.E. degree with the School of Optics and Photonics.

Her research interests include infrared object detection and tracking.