remote sensing

MDPI

Article

Cross-Parallel Attention and Efficient Match Transformer for Aerial Tracking

Anping Deng [1,2], Guangliang Han [1,*], Zhongbo Zhang [3], Dianbing Chen [1], Tianjiao Ma [1] and Zhichao Liu [1,2]

[1] Changchun Institute of Optics, Fine Mechanics and Physics (CIOMP), Chinese Academy of Sciences, Changchun 130033, China; denganping20@mails.ucas.ac.cn (A.D.); chendb@ciomp.ac.cn (D.C.); matianjiao@ciomp.ac.cn (T.M.); liuzhichao201@mails.ucas.ac.cn (Z.L.)
[2] University of Chinese Academy of Sciences, Beijing 101408, China
[3] School of Mathematics, Jilin University, Changchun 130012, China; zhongbozhang@jlu.edu.cn
* Correspondence: hangl@ciomp.ac.cn

Abstract: Visual object tracking is a key technology that is used in unmanned aerial vehicles (UAVs) to achieve autonomous navigation. In recent years, with the rapid development of deep learning, tracking algorithms based on Siamese neural networks have received widespread attention. However, because of complex and diverse tracking scenarios, as well as limited computational resources, most existing tracking algorithms struggle to ensure real-time stable operation while improving tracking performance. Therefore, studying efficient and fast-tracking frameworks, and enhancing the ability of algorithms to respond to complex scenarios has become crucial. Therefore, this paper proposes a cross-parallel attention and efficient match transformer for aerial tracking (SiamEMT). Firstly, we carefully designed the cross-parallel attention mechanism to encode global feature information and to achieve cross-dimensional interaction and feature correlation aggregation via parallel branches, highlighting feature saliency and reducing global redundancy information, as well as improving the tracking algorithm's ability to distinguish between targets and backgrounds. Meanwhile, we implemented an efficient match transformer to achieve feature matching. This network utilizes parallel, lightweight, multi-head attention mechanisms to pass template information to the search region features, better matching the global similarity between the template and search regions, and improving the algorithm's ability to perceive target location and feature information. Experiments on multiple drone public benchmark tests verified the accuracy and robustness of the proposed tracker in drone tracking scenarios. In addition, on the embedded artificial intelligence (AI) platform AGX Xavier, our algorithm achieved real-time tracking speed, indicating that our algorithm can be effectively applied to UAV tracking scenarios.

Keywords: visual object tracking; UAV tracking; efficient match transformer; attention method

Citation: Deng, A.; Han, G.; Zhang, Z.; Chen, D.; Ma, T.; Liu, Z. Cross-Parallel Attention and Efficient Match Transformer for Aerial Tracking. *Remote Sens.* **2024**, *16*, 961. https://doi.org/10.3390/rs16060961

Academic Editor: Gemine Vivone

Received: 16 September 2023
Revised: 28 December 2023
Accepted: 6 January 2024
Published: 9 March 2024

1. Introduction

Visual object tracking is a fundamental research topic in the fields of computer vision and pattern recognition [1]. Its core task is to estimate the trajectory and state of an object continuously and stably in subsequent video sequences, given the initial position and scale information of the object. In recent years, due to the high maneuverability and payload capacity of unmanned aerial vehicle (UAV) platforms, increasing attention has been paid to the applications of drone platforms, such as video surveillance, human–computer interaction, aerial reconnaissance, and military strikes [2]. Visual object tracking technology has become a research hotspot in these areas. Benefiting from the development of deep learning theory and embedded platforms, many excellent drone-based tracking algorithms have emerged [3]. However, in practical engineering applications, the algorithms must contend with complex scenarios such as changes in target states, background interference causing feature loss, and environmental noise interference. In these scenarios, the tracking

accuracy and robustness of the algorithms are tested. Even more demanding is the fact that multiple challenging factors may appear simultaneously. Another thing to note is that embedded platforms have limited computational resources, and arbitrarily stacking various methods to improve accuracy can result in a high computational load. Carefully designing optimization methods targeted at embedded platforms is necessary to ensure lightweight algorithms. Therefore, determining how to design an accurate and robust object tracking algorithm that meets the real-time requirements of drone platforms remains an urgent problem to be solved.

In practical engineering applications, correlation filter–based methods have become quite advanced, thus most UAV platforms adopt correlation filter–based methods for aerial tracking [4]. However, as a result of complex optimization strategies and low-quality human-designed features, correlation filter–based tracking lacks robustness and is prone to tracking failure when confronted with various complex challenges [5]. On the other hand, Siamese trackers based on deep learning have achieved a good balance between accuracy and efficiency. Additionally, with the rapid development of deep learning technology and embedded processor computing power in recent years, it has become possible to deploy excellent tracking algorithms in real-time on drones. This has meant that the deployment of lightweight, real-time, deep learning–based trackers on UAVs has become a research hotspot. However, the existing algorithms still focus on improving accuracy through various serial units, ignoring the excellent parallel computing abilities of embedded platforms.

Therefore, this paper proposes a cross-parallel attention and efficient match transformer for aerial tracking, as shown in Figure 1. We introduce cross-parallel attention to extract efficient channel descriptors via parallel sub-networks to enhance feature saliency, and an efficient match transformer to aggregate and match the template and search region features for better feature matching.

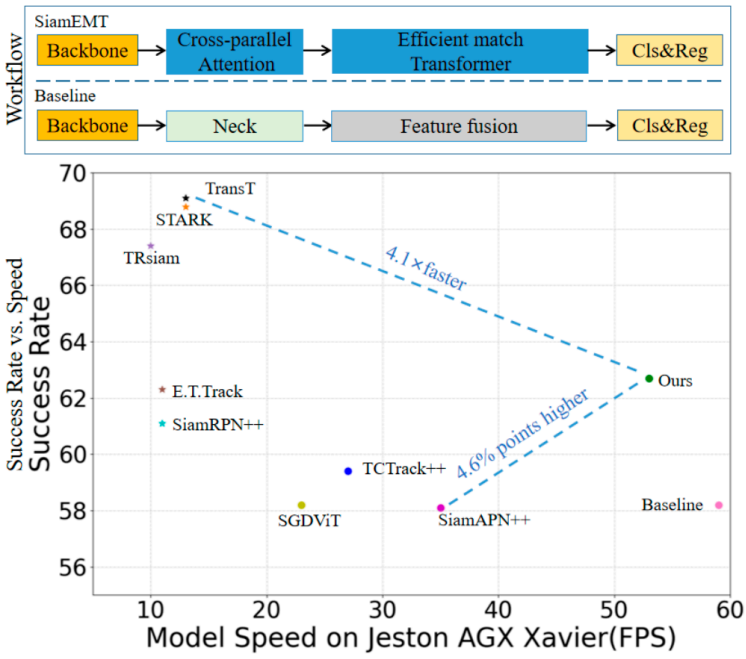

Figure 1. Comparison between the proposed tracker and other state-of-the-art (SOTA) methods. The figures, from top to bottom, are Workflow and Success Rate vs. Speed. The workflow diagram compares the cross-parallel attention and efficient match transformer for aerial tracking (SiamEMT) to the baseline by highlighting the differences: the introduction of a cross-parallel attention and efficient match transformer. In the Success Rate vs. Speed graph, it can be seen that SiamEMT has a 4.6% higher area under curve (AUC) score than SiamAPN++, and our frames per second (FPS) are 4.1 times faster than the transformer-based algorithm TransT. Compared with the baseline, this paper sacrifices the operation speed of 5 fps for a 4.5% accuracy improvement.

The main contributions of this paper can be summarized as follows:

- We propose a novel matching network called an efficient match transformer, which learns and matches features between search regions and effectively targets templates through parallel lightweight multi-head attention. This approach results in more reliable response maps.
- We design a cross-parallel attention mechanism that interacts and selectively aggregates multi-dimensional information from three parallel computational paths within a parallel sub-network, enhancing the algorithm's ability to discriminate between foreground and background.
- Extensive experiments and ablation studies have been conducted on multiple challenging public benchmarks, and SiamEMT has been proven to perform better in various complex scenarios through qualitative analysis and accuracy evaluations in complex scenarios. Additionally, the computational efficiency of our algorithm has been demonstrated through speed tests, with SiamEMT achieving 190 FPS on a PC and 53 FPS on an NVIDIA Jeston AGX Xavier, meeting the real-time requirements of engineering applications. The aforementioned experiments have validated the effectiveness and advanced nature of the tracker.

2. Related Works

The significant effectiveness of channel or spatial attention mechanisms in generating clearer feature representations has been demonstrated in various computer vision tasks. CBAM [6] algorithms infer attention maps along two independent dimensions (channel and space) in sequence, and then multiply the attention map with the input feature map for adaptive feature optimization. However, this algorithm introduces a large amount of computation and parameter requirements. Determining how to improve feature saliency while ensuring lightweight computing has become a research hotspot. ECANet [7] proposes a non-dimensionality reduction and local cross-channel interaction strategy, along with an adaptive method for selecting the size of one-dimensional convolution kernels to determine the coverage of local cross-channel interactions. EMA [8] re-examines the coordinate attention block, encodes global information to recalibrate channel weights in each parallel branch, and further aggregates the output features of two parallel branches through cross-dimensional interaction to capture pixel-level pairwise relationships. SiamITL [9] uses a three-branch structure to capture cross-dimensional interactions to calculate attention weights, establishes inter-dimensional dependencies through rotation operations and residual transformations, and encodes inter-channel and spatial information with negligible computational overhead. Taking inspiration from the above lightweight algorithms, we hope to fully leverage the efficient and lightweight features of parallel attention mechanisms to further improve the accuracy of UAV object tracking algorithms.

Visual object tracking is the key to enabling drones to perform reconnaissance, positioning, and strike missions for targets on the ground, in the air, and at sea. In recent years, scholars have continuously explored object tracking technology, resulting in a large number of visual object tracking algorithms [10]. These algorithms can be divided into two categories: correlation filter–based algorithms and deep learning–based algorithms.

The MOSSE [11] algorithm pioneered the correlation filter tracking framework through the online learning of equations and grayscale features. In subsequent improvements, the KCF [12] algorithm utilizes multi-channel directional gradient histograms to represent the target, introduces a circulant matrix to enhance classifier quality, and utilizes a Gaussian kernel function to transform low-dimensional nonlinear problems into high-dimensional linear problems. Through these methods, the computational complexity of the algorithm is significantly reduced. The KCF algorithm is a classic object tracking method widely used for object tracking tasks in UAVs. However, the limited feature representation ability of the manually designed features limits the tracking accuracy and robustness of correlation filter algorithms, making it difficult for them to effectively cope with the complex scenarios frequently encountered in object tracking tasks for UAVs. With the development of deep

learning technology, correlation filter algorithms have also introduced convolutional neural networks to extract deep features. UPDT [13] adaptively fuses deep features and manually designed features for tracking, and AutoTrack [14] fully utilizes local and global information in response maps to automatically adjust the hyperparameters of spatiotemporal regularizers. However, these methods require fine-tuning network models for different tracking tasks, making it difficult to ensure accuracy and generalization ability.

The Siamese neural network has received attention because of its end-to-end nature, with its algorithm process involving the identification of the most similar regions within the search area to the target template and estimating the target's state. The SiamFC [15] network pioneered the end-to-end Siamese neural network framework, transferring the object tracking problem from a similarity-matching problem to a classification problem between the foreground and background. Subsequent optimization algorithms have improved the robustness of the algorithm through various strategies such as auxiliary branches, multi-scale information fusion, template update mechanisms, and loss functions. The accuracy of feature matching plays a crucial role in the accuracy of tracking algorithms. PGNet [16] is a pixel-to-global feature-matching method that obtains more robust response map information through a combination of multiple feature-matching methods. SiamGAT [17] improves non-local attention with a graph neural network to facilitate similarity learning. TransT [18] introduces a large transformer [19] algorithm for global feature-matching combination modules and search area feature information. AiATrack [20] proposes a new feature-extraction and information-transmission module that solves some of the problems related to noise, blurred weights, and interference from similar objects in the response map introduced by key-value and query computations performed separately. However, the feature-matching process for these methods is relatively complex and has significant computational requirements that can be difficult to run on embedded platforms effectively and efficiently.

With the widespread application of drone technology, the demand for real-time visual tracking algorithms is constantly increasing. TCTrack++ [21] integrates temporal information at both the feature and response map dimensions and designs a novel training strategy to refine feature information. SGDiViT [22] introduces a saliency-guided dynamic vision transformer to refine the feature-matching process and enhance the algorithm's focus on appearance information. SiamSTM [23] designs a lightweight multiple matching network that fuses target focus information and effectively utilizes focus information to assist the feature-matching process. SiamTPN [24] utilizes the feature pyramid structure of lightweight networks and maximizes the enhancement of multi-scale target feature representation through transformer architecture. Although the aforementioned algorithms stack serial modules to increase network depth, they fail to establish information correlation between the two branches, leaving further room for improvement in terms of performance and runtime speed.

3. Proposed Method

In this section, we introduce the composition of SiamEMT and the details of the cross-parallel attention and efficient match transformer. The framework of SiamEMT is shown in Figure 2, and is divided into input, feature extraction, feature fusion, target localization, and output.

3.1. Overall Overview

During the input stage, we select a square region twice the size of the initial target bounding box as the template region ($Z \in R^{127\times127\times3}$), and a square region four times the size of the previous frame's predicted bounding box as the search region ($X \in R^{255\times255\times3}$). These two branches are sent into the feature extraction stage. We use a modified mobilenetV2 [25] as the feature extraction network. Specifically, we modify the stride from 8 to 16, because a larger stride can quickly reduce the resolution of the feature map, reducing the computational and parameter requirements of the network. Subsequently, the

two branches' features are sent into cross-parallel attention to enhance their saliency and discriminability, resulting in two groups of feature vectors: template ($Z \in R^{8\times8\times96}$) and search region ($X \in R^{16\times16\times96}$). Parameter sharing refers to the fact that the two branches' backbones and cross-parallel attention share parameters in the network, ensuring algorithm coherence without additional training. The feature-matching process is implemented in our carefully designed efficient match transformer. First, we interact the features of the two branches through parallel lightweight multi-head attention to obtain more accurate features in parallel operations. The feature match transformer is then used to complete feature matching and obtain a response map ($F\in R^{16\times16\times64}$). Finally, the response map is sent to a target location to obtain target state information through a classification and regression network, resulting in the output.

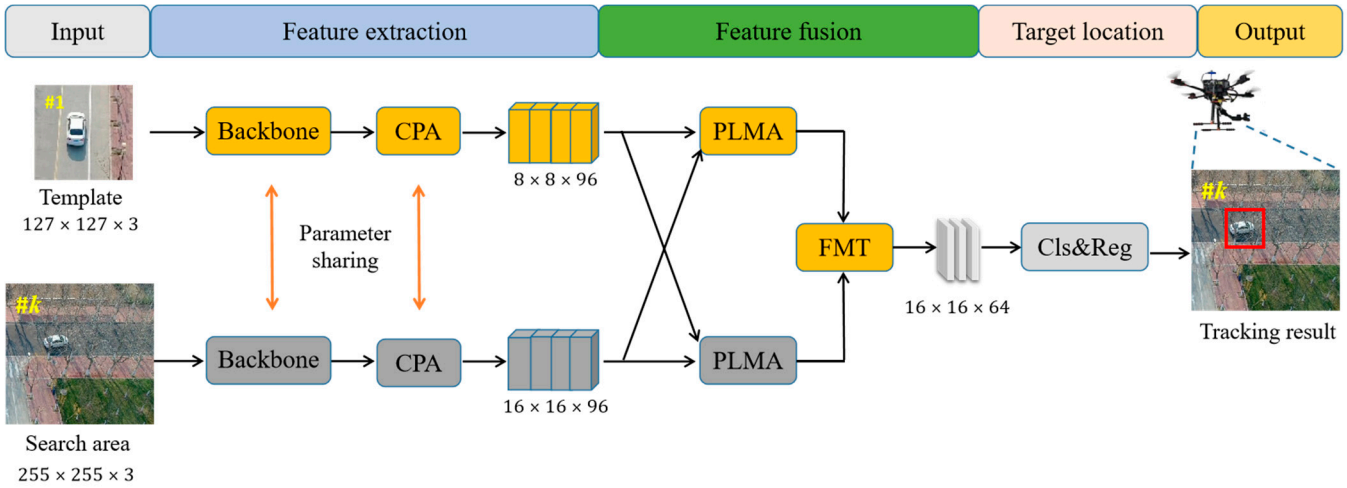

Figure 2. The overall framework of SiamEMT. CPA means cross-parallel attention, and PLMA and FMT refer to parallel lightweight multi-head attention and feature match transformer. The amounts 255 × 255 × 3 represents height × width × channel of the characteristic graph, respectively. Compared with the baseline, we added the CPA, PLMA, and FMT modules, as well as other modules to be as consistent as possible with the baseline.

3.2. Cross-Parallel Attention

Thanks to the powerful parallel computing capabilities of gpus and embedded platforms, the computational burden is no longer a bottleneck in speed. Minimizing the number of serial layers is more effective than reducing computation, and parallel sub-network structures help the network introduce various feature-enhancement modules while maintaining computational speed. In this section, we discuss how the cross-parallel attention mechanism can extract efficient channel descriptors through three parallel branches to enhance feature saliency. The overall structure of the cross-parallel attention mechanism is shown in Figure 3.

The local receptive field in convolution allows the network to acquire valuable spatial context information. Therefore, cross-parallel attention interacts with multi-dimensional information through three parallel computational paths to capture the extent of influence between all channels, selectively aggregating feature context information from multiple attention coefficients. Specifically, we designed two 1 × 1 paths, one 3 × 3 path, and one residual connection. The two 1 × 1 paths encode feature information along two spatial directions, while the 3 × 3 path enhances the local neighborhood information of the algorithm. The residual connection maintains feature stability to prevent model degradation.

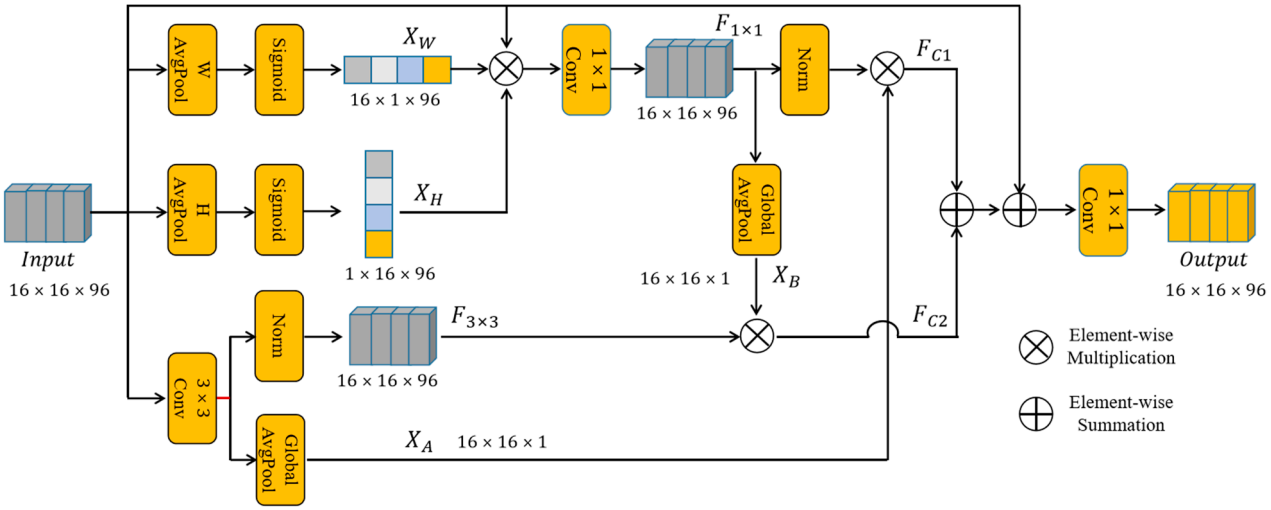

Figure 3. Structure of the cross-parallel attention mechanism. For ease of reading, we introduced names similar to X_W to distinguish the various feature maps.

In the first branch, we perform average pooling of the features along the width dimension and fit them to the feature weight coefficients on the width dimension using a nonlinear sigmoid function, obtaining the X_W. Similarly, in the second branch, we perform average pooling of the features along the height dimension and fit them to the feature weight coefficients on the width dimension using a nonlinear sigmoid function, obtaining the X_H. To aggregate the feature information along both width and height dimensions, we simply multiply the two attention weight coefficients with the input features to obtain the spatial attention weight coefficients, and use 1×1 convolution to enhance the nonlinearity of the features for better fitting and generalization capabilities, obtaining the output $F_{1\times1}$ of the 1×1 branch. In the 3×3 branch, we use 3×3 convolution and normalization to achieve local spatial information enhancement.

Next, we introduce the method for cross-dimensional information aggregation. On one hand, we obtain the global feature descriptor X_B, which collects spatial information from different dimensions through global average pooling, then multiply it with the feature map $F_{3\times3}$, resulting in F_{C2}. On the other hand, in the 3×3 branch, we use global average pooling to encode the local spatial information of the neighborhood, obtaining the local feature descriptor X_A and multiplying it with the feature map $F_{1\times1}$, resulting in F_{C1}. Finally, we sum the pixel-level values of the F_{C1}, F_{C2} residual connection input, and use the 1×1 convolution to achieve channel dimension reduction, allowing the network to learn the weights of each branch and adaptively capture global and local context information.

As mentioned earlier, cross-parallel attention obtains global and local descriptors through three parallel branches and aggregates them through cross-information aggregation. While modeling global contextual information, it enhances the algorithm's ability to perceive local neighborhood information. Finally, it aggregates information from each branch through a residual connection, while retaining the original feature information, selectively aggregating contextual information. The final output feature has overall spatial contextual information and exhibits better discriminant ability between the target of interest and background, improving the algorithm's ability to distinguish between the foreground and background.

3.3. Effcient Match Transformer

Object tracking algorithms rely on similarity matching to obtain the most likely location of the target within the search area. On one hand, most existing trackers use a deep correlation operator to accomplish similarity matching, which has no learnable parameters and matching results that rely entirely on the feature representation, making the algorithm

prone to losing semantic information [26]. On the other hand, inspired by the transformer, some advanced trackers adopt the transformer to achieve the feature-matching process and obtain reliable similarity maps [27], but this type of method significantly increases computational complexity, making it difficult to ensure real-time tracking for UAVs.

Given this, and taking advantage of the characteristics of parallel networks, this paper proposes a new efficient match transformer that aggregates feature information between the target template and the search area through parallel lightweight multi-head attention, and completes feature matching through a feature match transformer. In parallel lightweight multi-head attention, we replace the self-attention mechanism with parallel mutual-attention mechanisms that interactively process information from the target template and search area branches, preserving a wealth of background information that can effectively distinguish the target from the background. Its structure is shown in Figure 4.

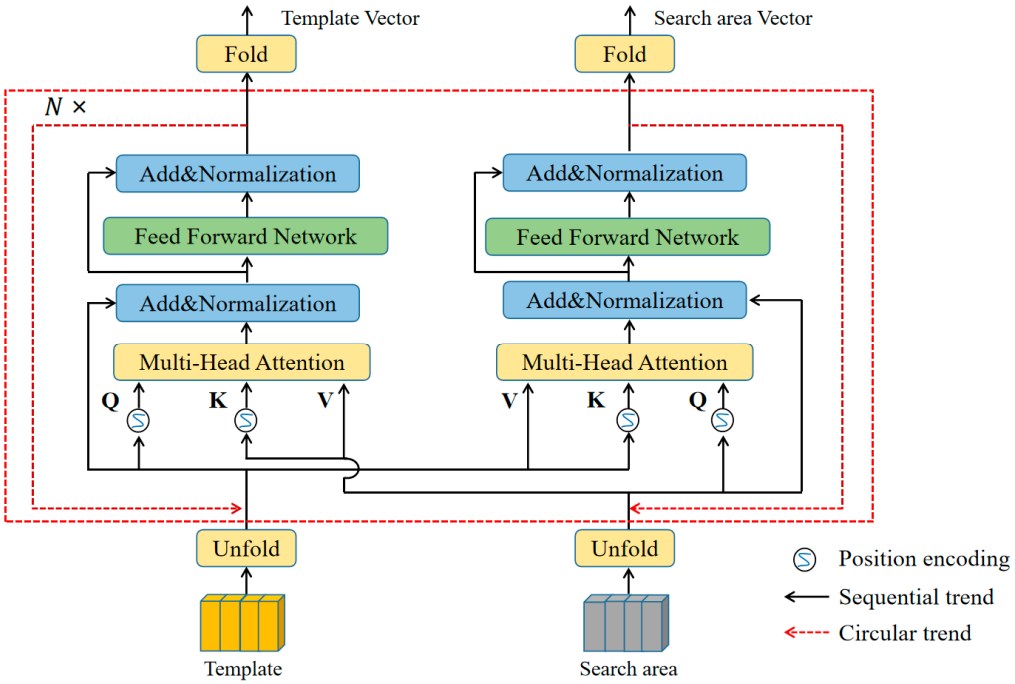

Figure 4. The schema of our parallel lightweight multi-head attention mechanism. Q, K, and V represent the input query, key, and value of the transformer structure, respectively. This structure has three cycles, N = 3.

In parallel lightweight multi-head attention, we propose a new interval attention approach for unfolding and folding features, as shown in Figure 4. Conventional attention flattens the feature map into a sequence and calculates the attention between each patch and all patches in the sequence. This computation method brings a significant amount of parameters and computations when stacking parallel transformer structures. Therefore, we adopt interval attention for the attention calculation, as shown in Figure 5, and we use the unfold module to divide the feature into four sequences, with the individual patches not interacting with their surrounding neighbors. We calculate the attention for each of these four sequences, resulting in a computational cost that is one quarter of that of the traditional attention approach. By using interval attention, we enhance the algorithm's ability to perceive global context while minimizing computational complexity.

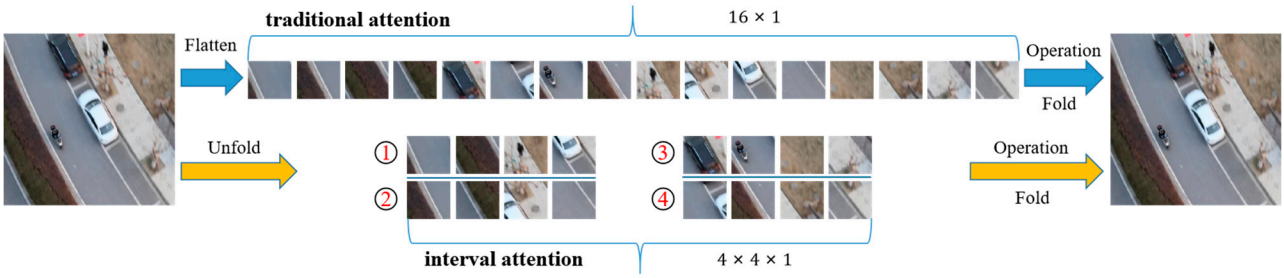

Figure 5. Structure of the traditional attention and interval attention mechanisms. Traditional attention divides images into a complete sequence, while interval attention divides images into four sequences.

The Template and Search areas are unfolded and serve as Q for one branch, and K and V for the other branch in the Multi-Head Attention. Next, Add&Layer Normalization is used to normalize the features and maintain their stability. The Feed Forward Network (FFN) consists of two fully connected neural networks, and it is worth noting that we adopt the GeLU function [28] as the activation function to generate a stronger nonlinear response and to obtain more generalized feature information. The output of the second layer Add&Layer Normalization is fed into another branch of Multi-Head Attention, and the sequential trend calculation result is obtained. The above process can be represented as:

$$\begin{aligned} \mathrm{PLMA}(\mathrm{Q},\mathrm{K},\mathrm{V}) &= \mathrm{Norm}(\mathrm{X}+\mathrm{FFN}(\mathrm{X})) \\ \mathrm{X} &= \mathrm{Norm}(\mathrm{Q}+\mathrm{MHA}(\mathrm{Q}+\mathrm{P_Q},\ \mathrm{K}+\mathrm{P_K},\ \mathrm{V})) \\ \mathrm{MHA}(\mathrm{Q},\mathrm{K},\mathrm{V}) &= \mathrm{Softmax}\left(\mathrm{Q}\cdot\mathrm{K}^{\mathrm{T}}/\sqrt{\mathrm{d}}\right)\cdot\mathrm{V} \end{aligned} \tag{1}$$

where $\mathrm{P_Q}$ and$\mathrm{P_K}$ represent the position coding information of Q and the position coding information of K, respectively. In the Multi-Head Attention (MHA) formula, the variance of the weight distribution is reduced by dividing the root of the number of channels (d) to ensure the stability of the forward and reverse variance.

In the method proposed in this paper, Q serves as one branch of information, while K and V serve as another branch. The information from the two branches is interactively fused through MHA. We generate position encoding using a sine function and preserve the positional information of the features. After undergoing N cycles, the output is the fused feature Template Vector and Search area Vector. Parallel lightweight multi-head attention (PLMA) is formulated as:

$$\begin{aligned} \mathrm{F}^{\mathrm{N}}_{\mathrm{Temp}} &= \mathrm{PLMA}\left(\mathrm{F}^{\mathrm{N}-1}_{\mathrm{Temp}},\ \mathrm{F}^{\mathrm{N}-1}_{\mathrm{Search}},\ \mathrm{F}^{\mathrm{N}-1}_{\mathrm{Search}}\right) \\ \mathrm{F}^{\mathrm{N}}_{\mathrm{Search}} &= \mathrm{PLMA}\left(\mathrm{F}^{\mathrm{N}-1}_{\mathrm{Search}},\ \mathrm{F}^{\mathrm{N}-1}_{\mathrm{Temp}},\ \mathrm{F}^{\mathrm{N}-1}_{\mathrm{Temp}}\right) \end{aligned} \tag{2}$$

where N represents the current layer number, and $\mathrm{N}-1$ denotes the output result of the previous loop. Through parallel lightweight multi-head attention, we establish a new parallel information interaction method between the Template branch and the Search area branch. For the Search area branch, this method enables the branch to learn the characteristics of the target, allowing for more effective discrimination between the target and background information, generating a stronger feature expression. In complex scenarios that involve target occlusion and similar target interference, the algorithm is able to identify the target more accurately. For the Template branch, this method enables the branch to learn the current frame's target state, achieving adaptive implicit template update, improving the discrimination ability in subsequent feature-matching processes for target status. After obtaining the enhanced Template Vector and Search area Vector through parallel lightweight multi-head attention, this algorithm completes the feature-matching process through the feature match transformer. The structure is shown in Figure 6.

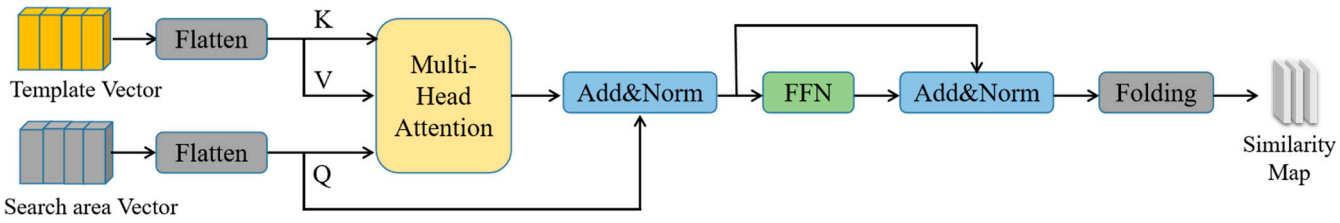

Figure 6. Architecture of our feature match transformer. To calculate the correlation between all patches and to obtain accurate matching results, we use the traditional attention calculation method here.

We establish a feature fusion network through the feature match transformer to establish a connection between the features of the two branches of the network. Specifically, we take Template as input K and V, Search area as input V, and obtain the most similar feature and position information of the template in the search area through MHA in Formula (1). It is worth noting that in this structure, we directly flatten the Template Vector and Search area Vector instead of using Fold for sorting. This is because the feature-matching process requires a global matching search strategy to obtain accurate classification results between the foreground and background, as well as to identify the precise location of the target. Unlike the standard vision transformer calculation process, we adopt a three-layer fully connected network in the FFN, which is connected by two GeLU layers. This is because this paper adopts multiple parallel transformers for linear feature fitting, which has poor generalization ability. Therefore, it is necessary to inject nonlinear factors through multiple fully connected network layers and excellent activation functions to ensure that the response map is closer to the target state.

4. Experiments

4.1. Implementation Details

In this paper, we adopt SiamRPN_MobileNetV2 [29] as our baseline algorithm because it exhibits balanced performance and efficiency. The prediction head consists of a classification network and a regression network, both of which comprise six prediction layers, with each layer containing a 3×3 block and a 1×1 block. The 3×3 block contains a 3×3 depthwise separable convolution, a BatchNormalization layer, and a GeLU activation function. The block also contains a depthwise separable convolution, a BatchNormalization layer, and a GeLU activation function. The backbone weights of the algorithm are initialized using ImageNet pre-trained weights, while the remaining weights are initialized using Kaiming initialization. To compare with Baseline more fairly, we adopted the same loss function as Baseline; that is, the classification loss uses the cross-entropy loss function, and the position loss uses the IOU loss.

The experimental equipment configuration included an Intel i7-12700 CPU, an NVIDIA RTX 3090 GPU, and 64 GB RAM. The embedded device used was a Nvidia Jeston AGX Xavier, with the Ubuntu18.04 software version. The training set for this article was derived from the LaSOT, TrackingNet, and GOT-10K datasets, and the size of the sampled search region was 255×255 The size of the target template was 126×126. During training, a random gradient descent with a momentum of 0.9 was used as the optimization strategy, and a warm-up training was applied for the first 5 epochs. A total of 20 epochs were trained, with each epoch containing 80,000 sampling pairs. The training duration was 6 h. To ensure parity, each version of the tracker used the same training strategy, hyperparameters, and testing environment. Credit goes to PySOT for providing a concise and clear testing environment.

4.2. Evaluation Index

The various testing benchmarks in the field of object tracking typically use accuracy and success rate as evaluation metrics. In Figure 7, we visually demonstrate the fundamentals of two evaluation metrics: center localization error and overlap score.

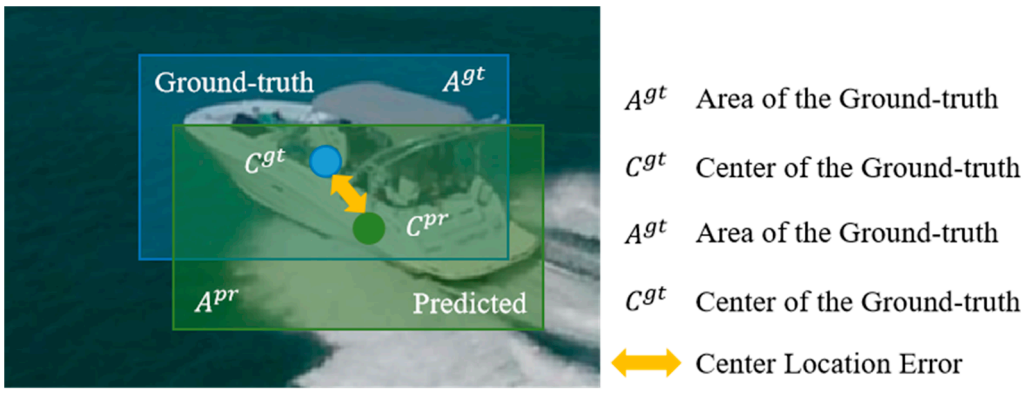

Figure 7. Schematic diagram of evaluation indicators.

The center localization error (CLE) in the figure refers to the Euclidean distance between the center of the ground truth bounding box (C^{gt}) and the center of the predicted bounding box (C^{pr}). Accuracy refers to the ratio of frames where the central localization error is less than a preset threshold of all frames. In the accuracy curve plot, the threshold is arranged as the horizontal coordinate from 0 to 50, and the vertical coordinate represents the tracking algorithm accuracy under the current threshold. It is typically considered that tracking is correct when the central localization error is less than 20 pixels, and its value is used as the accuracy of the algorithm.

The overlap score is the numerical value of the intersection over the union between the ground-truth bounding box region (A^{gt}) and the predicted bounding box region (A^{pr}). Success rate refers to the ratio of frames where the overlap score is less than a preset threshold of all frames. In the success rate curve plot, the threshold is arranged as the horizontal coordinate from 0 to 1, and the vertical coordinate represents the tracking algorithm success rate under the current threshold. It is typically considered that tracking is correct when the overlap score is greater than 0.5, and its value is used as the success rate of the algorithm. To highlight the differences in tracking algorithm capabilities, the success rates of tracking algorithms are ranked based on the area under the curve.

4.3. Experiments on the UAV123 Benchmark

The UAV123 [30] dataset consists of image sequences captured by a drone platform, with a total of 123 subsets. The tracking objects include pedestrians, vehicles, and ships, which are commonly encountered in drone ground tracking missions. The dataset covers three application scenarios: drone-to-ground, drone-to-sea, and drone-to-air. Because of the diverse application scenarios, there is always relative motion between the targets and the drone, resulting in complex scenarios. The dataset contains a total of 12 challenge attributes, including viewpoint change (VC), occlusion (PO), scale variation (SV), and fast motion (FM). In this test, our algorithm was compared with the state-of-the-art algorithms for UAV tracking, including TCTrack++, SGDViT, HIFT [31], SiamAPN++ [32], and SiamAPN. To demonstrate the excellent tracking ability of the algorithm in this paper, we selected representative tracking algorithms based on the large network structures used in recent years: SiamPW [33], SiamRPN, and SiamDW [34].

Overall Evaluation: As shown in Figure 8, the success rate of the algorithm in this paper is 0.627, and the accuracy rate is 0.819. Compared with the excellent UAV tracking algorithm TCTrack++ used in recent years, the algorithm in this paper has a success rate that is 3.3% higher and an accuracy rate that is 4.1% higher. When compared with the SiamPW, which is based on large-scale networks, the success rate and accuracy rate of the algorithm in this paper are superior. This indicates that the algorithm in this paper has better tracking accuracy and robustness.

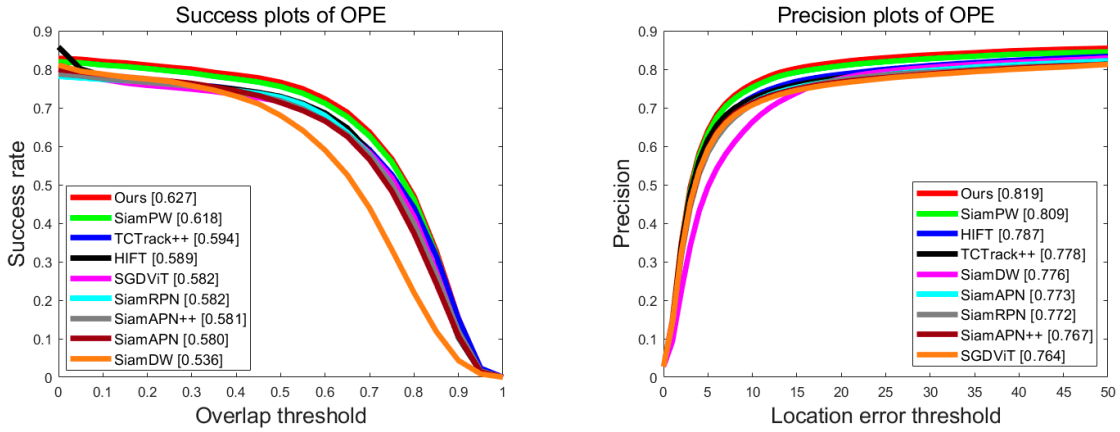

Figure 8. UAV123 comparison chart. The proposed algorithm SiamEMT performs favorably against state-of-the-art trackers. The left chart shows the success plots, and the right shows the precision plots.

Attribute-Based Evaluation: To fully compare the tracking algorithm's ability to handle various complex scenarios and to further demonstrate its robustness, we reveal the tracking accuracy of 12 complex scenarios from the UAV123 dataset: aspect ratio change (ARC), background clutter (BC), camera motion (CM), fast motion (FM), illumination variation (IV), low resolution (LR), out-of-view (OOV), similar object (SO), scale variation (SV), viewpoint change (VC), partial occlusion (PO), and full occlusion (FO). The success rate curve chart is shown in Figure 9, and the accuracy rate curve chart is shown in Figure 10.

The results in Figures 10 and 11 demonstrate that the algorithm proposed in this paper is not only superior in state-of-the-art UAV tracking, but also compares favorably with the large model–based tracker SiamPW. The algorithm achieves good tracking accuracy when encountering challenges such as changes in target state (CM, ARC, SV, VC, FM), environmental interference during the tracking process (BC, IV, LR, SO), and missing features caused by background interference (PO, FO, OOV). This indicates that the introduction of the efficient match transformer and cross-parallel attention has effectively improved the feature saliency by parallelizing attention to highlight target features and locations, enhanced the ability to perceive target appearance changes through parallel lightweight multi-head attention that interactively fuses features from the template branch and the search area branch, and reduced environmental noise interference using the feature match transformer to accurately locate target locations. The success rate and accuracy plots for the 12 challenge attributes effectively demonstrate the excellent tracking accuracy of the algorithm proposed in this this paper and its good robustness against complex scenarios.

4.4. Experiments on the UAV20L Benchmark

The UAV20L dataset consists of 20 long time sequences with an average sequence length of 3000 frames. The long tracking time of this benchmark results in cumulative target state changes, diverse and frequent environmental interference during the tracking process, and scenarios with missing features. The characteristics of this benchmark mean that the long-term tracking results are closer to the actual performance of tracking algorithms in UAV tracking. We compared our algorithm with the representative large-scale algorithms SiamBAN [35], SiamRPN++ [36], SiamCAR [37], SiamFC++ [38], and SESiamFC, as well as the excellent UAV tracking algorithms SGDViT, SiamAPN++, and SiamAPN.

Overall Evaluation: The success rate of the algorithm in long-term tracking scenarios is 0.593, and the accuracy rate is 0.764, as shown in Figure 11. Compared with the excellent UAV tracking algorithm SiamAPN++ used in recent years, the algorithm in this paper has a success rate that is 6.0% higher and an accuracy rate that is 6.1% higher. When compared with the representative large-scale networks SiamRPN++ and SiamFC++, the success rate and accuracy rate of the algorithm in this paper have a small lead of nearly 2%. This indicates that in long-term tracking scenarios, the tracking algorithm in this paper not only

accurately estimates the target state, but also achieves better judgment regarding the target center location.

Attribute-Based Evaluation: In Figure 12, we visually present the accuracy comparison between SiamEMT and the comparison algorithms in various complex scenarios using a radar chart. We selected eight scenarios that frequently occur in UAV tracking for comparison: low resolution, full occlusion, out-of-view, viewpoint change, scale variation, aspect ratio change, illumination variation, and camera motion.

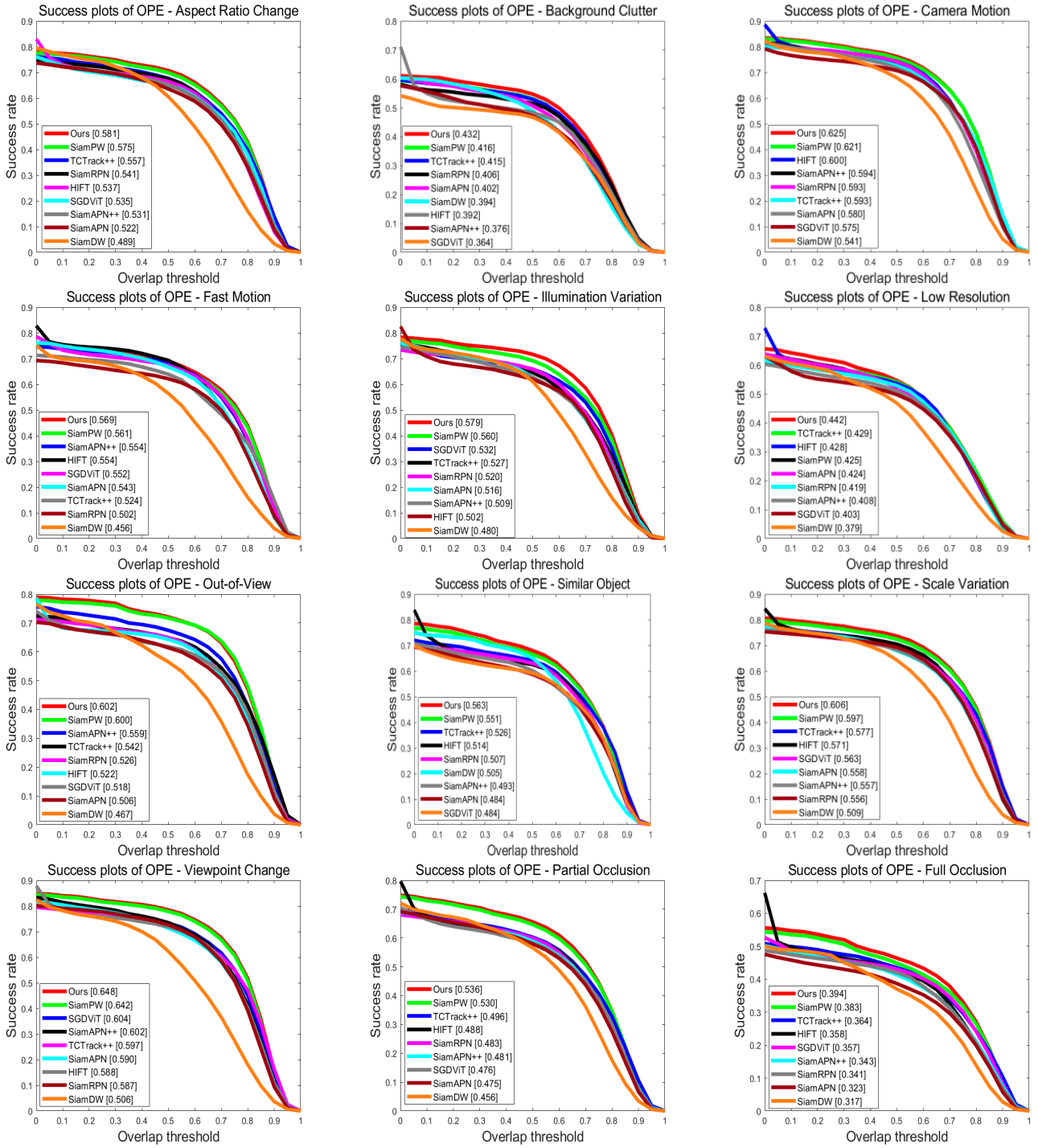

Figure 9. Success rates for different attributes of the UAV123 benchmark. Our tracker achieves superior performance against the other eight SOTA trackers.

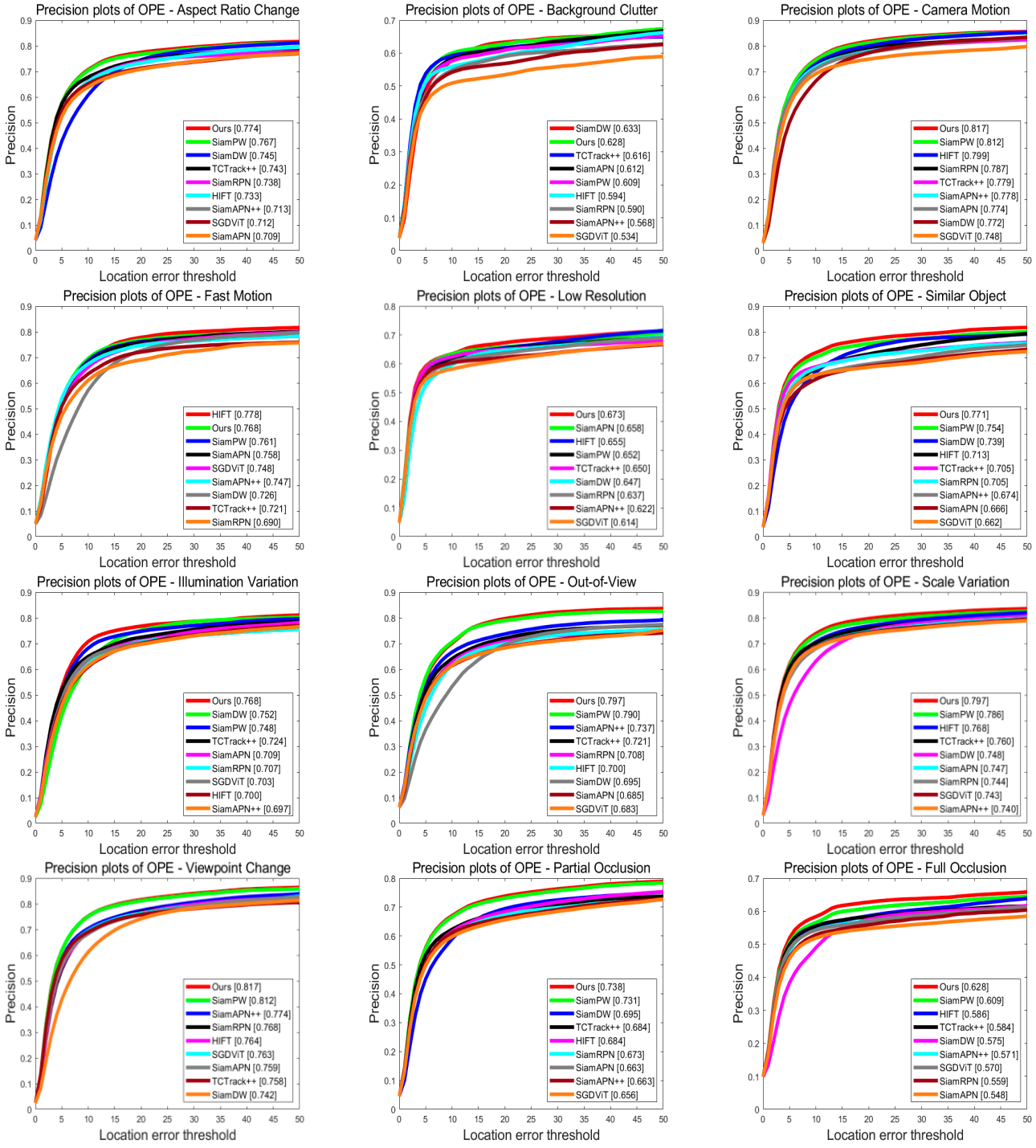

Figure 10. Precision plots for different attributes of the UAV123 benchmark. Our tracker achieves superior performance against the other eight SOTA trackers.

From the results shown in Figure 12, it can be seen that the algorithm proposed in this paper significantly outperforms the state-of-the-art algorithm in UAV tracking in all attributes. When compared with the algorithms based on large networks, the algorithm proposed in this paper has a certain degree of superiority in dealing with the challenges of target state changes (CM, ARC, SV, VC) and background interference that results in missing features (FO, OOV). However, when facing environmental interference during the tracking process (IV, LR), the success rate of the algorithm proposed in this paper is lower than that of the large network algorithms. This is because SiamEMT adopts a lightweight

feature extraction network. In addition, after enhancing feature saliency through cross-parallel attention and improving the feature-matching ability through the efficient match transformer, when external environments cause a decrease in feature resolution and clarity, the algorithm proposed in this paper will still be affected by the limitations of the feature extraction network to varying degrees. In a comprehensive evaluation of the eight complex scenarios, the overall accuracy and robustness of the algorithm proposed in this paper still maintain competitiveness compared with large network models.

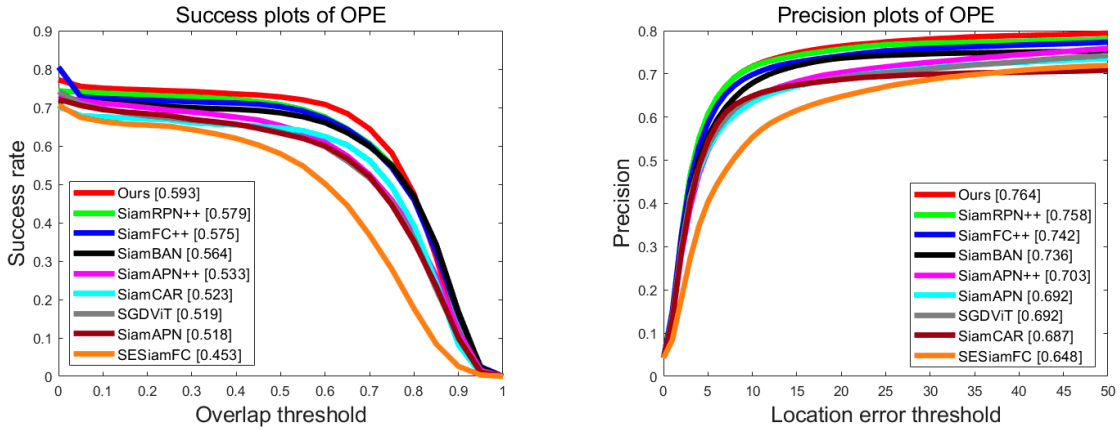

Figure 11. UAV20L comparison chart. The proposed algorithm SiamEMT performs favorably against state-of-the-art trackers. The left chart shows the success plots, and the right shows the precision plots.

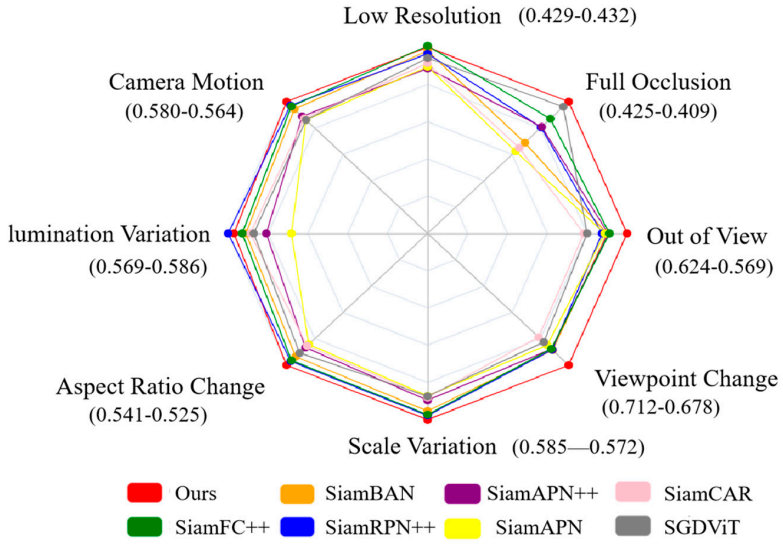

Figure 12. Success rates of different attributes on the UAV20L benchmark. The numbers in parentheses, such as (0.541–0.525), represent the accuracy of SiamEMT-the highest accuracy besides ours.

4.5. Comparison of Algorithm Efficiency

In this section, we divide the speed comparison into two parts: a comparison with state-of-the-art (SOTA) algorithms in UAV tracking and a comparison with large-scale network algorithms. The running speed of an algorithm is influenced by multiple factors, such as computational complexity, parameter count, memory footprint, and CUDA low-level support. We chose the most basic Model_Size and compared the running frame rates of the algorithm on various platforms. The DTB70 [39] dataset provides 70 short low-altitude UAV tracking sequences with an image resolution of 1280 × 720, which is a common image resolution for drone collections. The running speed of the algorithm on this benchmark is representative. We conducted tests on the DTB70 dataset, as shown in Table 1.

Table 1. Attribute-based evaluation of the SiamEMT and other six SOTA trackers on the DTB70 benchmark. ~ represents unpublished data.

Trackers	Backbone	Accuracy		Model_Size (MB)	FPS_GPU (FPS)	FPS_Xavier (FPS)
		Pre.	Suc.			
SiamAPN	AlexNet	0.784	0.859	118.7	180	34
LightTrack [40]	NAS	0.761	0.587	~	128	38
SiamAPN++	AlexNet	0.790	0.594	187.1	175	35
HiFT	AlexNet	0.802	0.594	82.1	128	31
SGDViT	AlexNet	0.806	0.603	183.2	116	23
TCTrack++	TC_AlexNet	0.813	0.626	113.0	~	27
Ours	MobilenetV2	0.817	0.631	71.2	190	53

Table 1 shows a comparison of the accuracy and speed of the algorithms. In terms of accuracy, the algorithm proposed in this paper is superior to the comparison algorithms in both success rate and accuracy rate. The Model_Size of the algorithm proposed in this paper is the smallest, indicating that the efficient match transformer and cross-parallel attention are introduced to ensure the simplicity and lightweightness of the algorithm. On the GPU platform, the algorithm proposed in this paper is the most efficient with 195 FPS, which is 1.68 times faster than the open-source algorithm SGDViT. On the embedded platform Xavier, the algorithm proposed in this paper achieves an excellent speed of 55 FPS, which is 2.03 times faster than the accuracy runner-up TCTrack++. To further demonstrate the effectiveness of the algorithm proposed in this paper, we compared it with advanced tracking algorithms based on large-scale network models, as shown in Table 2. The NFS [41] dataset samples images with a high frame rate, which imposes higher requirements on the running speed of the algorithm and requires a careful trade-off between computational cost and application accuracy. The GOT-10K [42] dataset has an image resolution of 1920 × 1080, and contains various targets and rich motion trajectory information, making it an authoritative and practical evaluation benchmark.

Table 2. State-of-the-art comparison of NFS, UAV123, and GOT-10K benchmarks. ~ represents unpublished data.

	Non-Realtime							Realtime	
	SiamRPN++	PrDiMP [43]	TrSiam [44]	TransT	STARK [45]	E.T.Track [46]	FEAR [47]	LightTrack	SiamEMT (Ours)
NFS	50.2	63.5	65.8	65.7	66.4	59.0	61.4	55.3	63.2
UAV123	61.1	68.0	67.4	69.1	68.8	62.3	~	62.5	62.7
GOT-10K	51.7	63.4	67.3	72.3	68.0	~	61.9	61.1	61.9
CPU Speed	4	6	5	5	7	47	60	41	43
GPU Speed	56	47	5	63	50	40	105	128	190
AGX Speed	11	11	5	13	13	20	38	38	53

Whether or not an algorithm can achieve a running speed of 30 FPS on the AGX Xavier was used as an indicator of whether it runs in real time. As shown in Table 2, although there is a gap in model accuracy between the algorithm proposed in this paper and large-scale models, the algorithm proposed in this paper can achieve real-time operation on embedded devices by sacrificing only a slight reduction in accuracy, and achieving a tracking speed that is multiple times faster. Compared with real-time algorithms, the algorithm proposed in this paper further optimizes the running speed of the tracking algorithm on embedded devices while improving accuracy. Overall, SiamEMT achieves fast and efficient operation on multiple devices, and we believe that parallel network structures are helpful for effective tracking in real time.

4.6. Qualitative Evaluation

To visually demonstrate the robustness of our method against various complex scenarios, we illustrate the tracking results with the ground-truth bounding box in blue, the predicted bounding box of our algorithm in red, and compare it with the SOTA algorithm TCTrack++ represented by the yellow bounding box, as displayed in Figure 13. We also objectively analyze and compare the tracking accuracy using CLE curves for each sequence.

For the first test sequence Car1_s, the car travels with turns, scale changes, and variations in lighting conditions caused by the scene. From frame 1 to frame 203, the car rapidly travels forward, causing a decrease in the target scale. The comparison algorithm TCTrack++ experiences fluctuations in tracking accuracy as a result of the scale changes. Around frame 386, a turn by the car introduces appearance changes that pose a challenge. The TCTrack++ algorithm does not match the target size well. From frame 869 to frame 941, the drone quickly rises and the lighting intensity undergoes significant changes. Because of these factors, the yellow bounding box of the comparison algorithm experiences tracking failure, while the algorithm proposed in this paper can overcome these effects and maintain stable tracking throughout the entire process.

For the Bike1 test sequence, the sequence is long and is accompanied by complex challenges, such as changes in target appearance and drone perspective, and interference from similar targets. From frame 1 to frame 1070, as a result of the constantly changing perspective of the drone, the appearance and size of the target also change, and it can be seen that TCTrack++ maintains a high error in the center position throughout this process and is constantly on the edge of tracking failure. Around frame 1881, the bicycle stops and turns, and both the algorithm proposed in this paper and the comparison algorithm are affected by the significant appearance changes to varying degrees. Around frame 2248, a similar target partially overlaps with the target, causing the tracking box of the comparison algorithm to also include the other target, while the algorithm proposed in this paper can maintain stable tracking of the target.

The Truck2 sequence is a classic drone ground tracking scenario where the tracking object, a truck, is occluded by a building. Throughout the entire sequence, the comparison algorithm experiences tracking failure because of the occlusion. When the target is completely occluded, the comparison algorithm remains in the area where the target is lost and fails to capture the reappearance of the target, resulting in the tracking failure scenario in frame 181 and the tracking drift scenario where another truck is falsely tracked in frame 216. However, the algorithm proposed in this paper benefits from the efficient match transformer's ability to effectively capture information in the search area and maintains good tracking accuracy throughout the entire sequence.

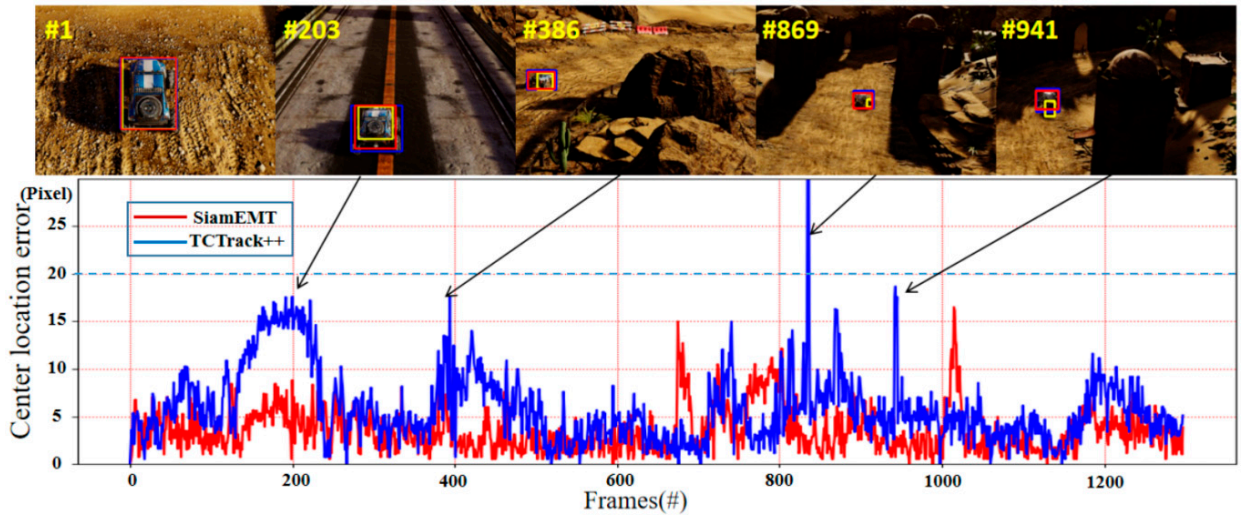

Figure 13. *Cont.*

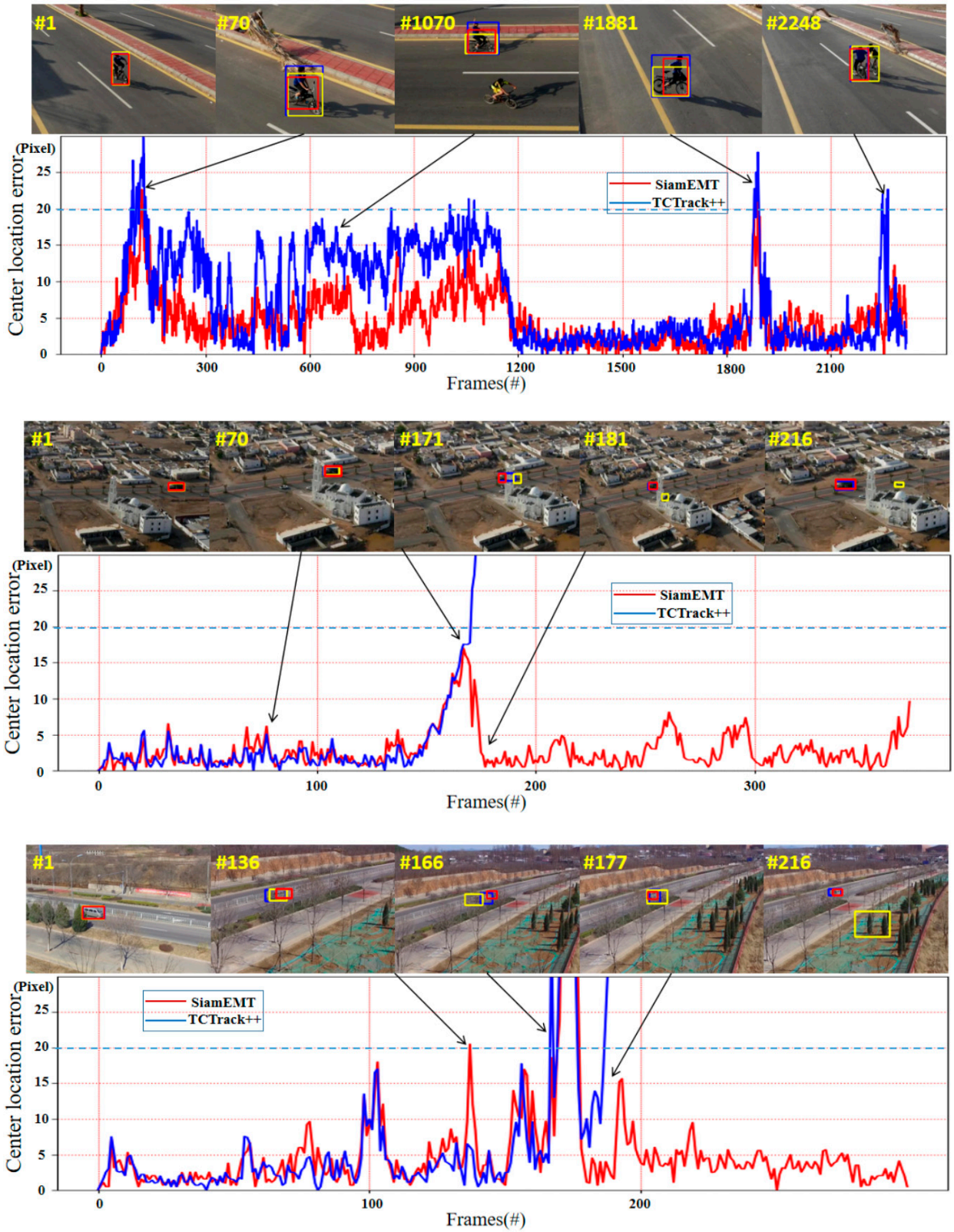

Figure 13. *Cont.*

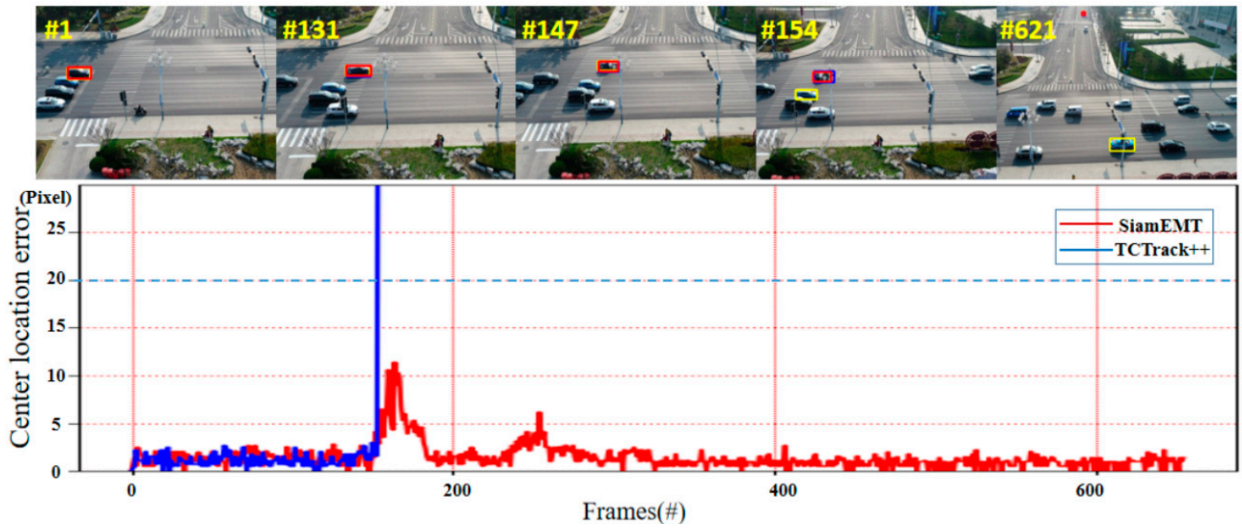

Figure 13. Success rates of different attributes on the UAV123 and UAVDT benchmarks. From top to bottom: Car1_s, Bike1, and Truck2 sequences from the UAV123 dataset; and S1101 and S1606 from the UAVDT benchmark. The yellow frame represents the baseline, the blue frame represents ground truth, and the red frame represents our method. The CLE below the purple dashed line represents the success tracking in the test. The first frame of the sequence displays the initial state of the object. To better display the tracking box, we cropped and enlarged the image.

In the S1101 sequence, the drone changes its perspective, but there is no relative movement, and the car quickly turns and drives towards the distance. From frame 136 to frame 216, because of the occlusion caused by trees, the comparison algorithm experiences significant accuracy fluctuations and loses track of the target around frame 216. However, the algorithm proposed in this paper, despite being influenced by occlusion and experiencing accuracy fluctuations, ultimately overcomes the complex scenarios and achieves stable tracking.

The S1606 sequence has a complex scene in which, from frame 1 to frame 154, the car is occluded by a street lamp while turning and there are similar objects in the surroundings. Under the interference of multiple factors, the comparison algorithm experiences tracking drift when it tracks a similar target at frame 154. However, the algorithm proposed in this paper maintains good tracking performance, and even at frame 621 when the car is driving towards the distance and is almost invisible in the image, it can still stably track the target.

The aforementioned qualitative experiments demonstrate that the algorithm proposed in this paper enhances feature saliency through cross-parallel attention, and improves the algorithm's ability to capture and perceive target status through the efficient match transformer, thereby effectively addressing the challenges posed by changes in target status (CM, ARC, SV, VC) and coping with the feature loss caused by background interference (FO, OOV). This is consistent with our conclusions in the quantitative analysis.

4.7. Ablation Study

To validate the structural effectiveness of the cross-parallel attention and efficient match transformer proposed in this paper, we present our ablation experiments in Table 3. For cross-parallel attention, this module enhances the feature information of the target, improves the success rate and accuracy of the algorithm, and does not have a significant effect on the tracking speed. After introducing the efficient match transformer, which completes the feature-matching process through parallel modules, the tracking accuracy of SiamEMT is further improved. Further, benefiting from the powerful parallel computing capabilities of the computing unit, the algorithm can achieve real-time tracking on all three platforms.

Table 3. Ablation study of the proposed tracker on the UAV123 benchmark. The √ and × indicate whether or not the module is in use, respectively. No. 1 represents the baseline, and when EMT is not added, traditional cross-correlation operations are used for target matching.

No.	CPA	EMT	PRE	AUC	CPU Speed	GPU Speed	AGX Speed
1	×	×	0.772	0.582	56	197	59
2	√	×	0.795	0.603	55	195	57
3	×	√	0.806	0.619	47	191	55
4	√	√	0.819	0.627	43	190	53

5. Discussion and Conclusions

In this work, we propose a cross-parallel attention and efficient match transformer (SiamEMT) tracking framework for aerial tracking. The framework is composed of two parts: cross-parallel attention and an efficient match transformer. Specifically, we utilize parallelized methods to avoid cumbersome serial computations and to alleviate computational resource consumption, thereby achieving a balance between the accuracy and speed of the tracking algorithm. Extensive experiments were conducted on multiple benchmarks and ablation studies to demonstrate the tracking accuracy of our method, and its feasibility in practical engineering applications was validated through speed tests on both a PC and the Xavier embedded platform.

We hope that our work can inspire the creation of more advanced UAV tracking algorithms. We believe that there are potential areas for improvement in our work. Limited by the capabilities of the feature extraction network, our algorithm still requires improvement in terms of dealing with environmental interference during the tracking process, for example, in long-term tracking tasks and scenarios where multiple complex challenges occur simultaneously. Moreover, with the help of TensorRT technology and the ONNX approach, the algorithm can be made to run faster on embedded devices. Our future work will focus on improving these two aspects and promoting the practical implementation of tracking-related applications. We will publish our code after completing the ONNX acceleration at https://github.com/Duranin/SiamEMT.

Author Contributions: All the authors participated in devising the tracking approach and made significant contributions to this work. A.D. devised the approach and performed the experiments; G.H. and D.C. provided advice for the preparation and revision of the work; T.M., Z.L. and Z.Z. assisted with the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the Department of Science and Technology of Jilin Province under Grant 20210201132GX of Dianbing Chen, and by the Science & Technology Development Project of Jilin Province, Key R&D Programs No. 20210201078GX of Zhongbo Zhang.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request. The data of the article can be obtained from our Github until we complete the deployment of onnx.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, F.; Wang, X.; Zhao, Y.; Lv, S.; Niu, X. Visual object tracking: A survey. *Comput. Vis. Image Underst.* **2022**, *222*, 103508. [CrossRef]
2. Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 91–124. [CrossRef]
3. Fu, C.; Lu, K.; Zheng, G.; Ye, J.; Cao, Z.; Li, B.; Lu, G. Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis. *arXiv* **2022**, arXiv:2205.04281. [CrossRef]
4. Fu, C.; Li, B.; Ding, F.; Lin, F.; Lu, G. Correlation filters for unmanned aerial vehicle-based aerial tracking: A review and experimental evaluation. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 125–160. [CrossRef]
5. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep Learning for Visual Tracking: A Comprehensive Survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 3943–3968. [CrossRef]

6. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
7. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
8. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
9. Deng, A.; Han, G.; Chen, D.; Ma, T.; Wei, X.; Liu, Z. Interframe Saliency Transformer and Lightweight Multidimensional Attention Network for Real-Time Unmanned Aerial Vehicle Tracking. *Remote Sens.* **2023**, *15*, 4249. [CrossRef]
10. Soleimanitaleb, Z.; Keyvanrad, M.A. Single object tracking: A survey of methods, datasets, and evaluation metrics. *arXiv* **2022**, arXiv:2201.13066.
11. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual Object Tracking Using Adaptive Correlation Filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
12. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
13. Bhat, G.; Johnander, J.; Danelljan, M.; Khan, F.S.; Felsberg, M. Unveiling the Power of Deep Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 483–498.
14. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Lu, G. AutoTrack: Towards High-Performance Visual Tracking for UAV With Automatic Spatio-Temporal Regularization. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11920–11929. [CrossRef]
15. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P. Fully-Convolutional Siamese Networks for Object Tracking. In *Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–16 October 2016*; Part II 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 850–865.
16. Wang, P.; Zhang, C.; Qi, F.; Liu, S.; Zhang, X.; Lyu, P.; Shi, G. Pgnet: Real-Time Arbitrarily-Shaped Text Spotting with Point Gathering Network. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 2782–2790.
17. Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; Shen, C. Graph Attention Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 9543–9552.
18. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 8126–8135.
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
20. Gao, S.; Zhou, C.; Ma, C.; Wang, X.; Yuan, J. Aiatrack: Attention in Attention for Transformer Visual Tracking. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–37 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 146–164.
21. Cao, Z.; Huang, Z.; Pan, L.; Zhang, S.; Liu, Z.; Fu, C. TCTrack: Temporal Contexts for Aerial Tracking. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 14778–14788. [CrossRef]
22. Yao, L.; Fu, C.; Li, S. SGDViT: Saliency-Guided Dynamic Vision Transformer for UAV Tracking. *arXiv* **2023**, arXiv:2303.04378.
23. Deng, A.; Han, G.; Chen, D.; Ma, T.; Liu, Z. Slight Aware Enhancement Transformer and Multiple Matching Network for Real-Time UAV Tracking. *Remote Sens.* **2023**, *15*, 2857. [CrossRef]
24. Xing, D.; Evangeliou, N.; Tsoukalas, A.; Tzes, A. Siamese Transformer Pyramid Networks for Real-Time UAV Tracking. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 4–8 January 2022; pp. 2139–2148.
25. Howard, A.; Zhmoginov, A.; Chen, L.C.; Sandler, M.; Zhu, M. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
26. Fu, Z.; Fu, Z.; Liu, Q.; Cai, W.; Wang, Y. Sparsett: Visual tracking with sparse transformers. *arXiv* **2022**, arXiv:2205.03776.
27. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. Swintrack: A simple and strong baseline for transformer tracking. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 16743–16754.
28. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
29. Bo, L.; Yan, J.; Wei, W.; Zheng, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
30. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the Computer Vision–ECCV 2016 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Part I 14. Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 445–461.

31. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. Hift: Hierarchical Feature Transformer for Aerial Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15457–15466.
32. Fu, C.; Cao, Z.; Li, Y.; Ye, J.; Feng, C. Siamese Anchor Proposal Network for High-Speed Aerial Tracking. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May 2021–5 June 2021; pp. 510–516.
33. Tang, F.; Ling, Q. Ranking-Based Siamese Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orléans, LO, USA, 19–24 June 2022; pp. 8741–8750.
34. Zhipeng, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019.
35. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese Box Adaptive Network for Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6668–6677.
36. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
37. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese Fully Convolutional Classification and Regression for Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6269–6277.
38. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. Siamfc++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12549–12556.
39. Li, S.; Yeung, D.Y. Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 1, p. 31.
40. Yan, B.; Peng, H.; Wu, K.; Wang, D.; Fu, J.; Lu, H. Lighttrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021.
41. Kiani Galoogahi, H.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. Need for Speed: A Benchmark for Higher Frame Rate Object Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1125–1134.
42. Lianghua, H.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577.
43. Martin, D.; Van Gool, L.; Timofte, R. Probabilistic Regression for Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7183–7192.
44. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer Meets Tracker: Exploiting Temporal Context for Robust Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 1571–1580.
45. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10448–10457.
46. Blatter, P.; Kanakis, M.; Danelljan, M.; Van Gool, L. Efficient Visual Tracking with Exemplar Transformers. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023.
47. Borsuk, V.; Vei, R.; Kupyn, O.; Martyniuk, T.; Krashenyi, I.; Matas, J. FEAR: Fast, Efficient, Accurate and Robust Visual Tracker. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–37 October 2022; Springer Nature: Cham, Switzerland, 2022.