electronics

MDPI

Article

SiamUT: Siamese Unsymmetrical Transformer-like Tracking

Lingyu Yang [1], Hao Zhou [1,*], Guowu Yuan [1], Mengen Xia [1], Dong Chen [1], Zhiliang Shi [2] and Enbang Chen [2]

[1] School of Information Science and Engineering, Yunnan University, Kunming 650504, China; proyang98@gmail.com (L.Y.)
[2] Kunming Enersun Technology Co., Ltd., Kunming 650504, China
* Correspondence: zhouhao@ynu.edu.cn

Abstract: Siamese networks have proven to be suitable for many computer vision tasks, including single object tracking. These trackers leverage the siamese structure to benefit from feature cross-correlation, which measures the similarity between a target template and the corresponding search region. However, the linear nature of the correlation operation leads to the loss of important semantic information and may result in suboptimal performance when faced with complex background interference or significant object deformations. In this paper, we introduce the Transformer structure, which has been successful in vision tasks, to enhance the siamese network's performance in challenging conditions. By incorporating self-attention and cross-attention mechanisms, we modify the original Transformer into an asymmetrical version that can focus on different regions of the feature map. This transformer-like fusion network enables more efficient and effective fusion procedures. Additionally, we introduce a two-layer output structure with decoupling prediction heads, improved loss functions, and window penalty post-processing. This design enhances the performance of both the classification and the regression branches. Extensive experiments conducted on large public datasets such as LaSOT, GOT-10k, and TrackingNet demonstrate that our proposed SiamUT tracker achieves state-of-the-art precision performance on most benchmark datasets.

Keywords: computer vision; object tracking; siamese; transformer

Citation: Yang, L.; Zhou, H.; Yuan, G.; Xia, M.; Chen, D.; Shi, Z.; Chen, E. SiamUT: Siamese Unsymmetrical Transformer-like Tracking. *Electronics* **2023**, *12*, 3133. https://doi.org/10.3390/electronics12143133

Academic Editor: Savvas A. Chatzichristofis

Received: 13 June 2023
Revised: 12 July 2023
Accepted: 17 July 2023
Published: 19 July 2023

1. Introduction

Visual object tracking is a fundamental task in computer vision. Research on visual tracking has received increasing attention during the past decade. With the development of artificial intelligence and modern neural networks, many new methods sprang up and helped researchers achieve lots of significant breakthroughs in this area. Although such great progress has been made recently, single-object tracking is still considered a rather challenging and complex problem. Single-object tracking aims to predict the location and outline of a moving target given in a certain frame of a video in advance. There are lots of challenges, such as deformation, occlusion, clutter, scale variation, and so on, which make it even more difficult to predict precisely [1,2].

Trackers with the siamese network structure have been widely applied to improving tracking accuracy. Based on the siamese structure, a large number of modified models occurred in order to decrease the side-effects of the linear computation in the siamese network. Correlation-based networks tend to fall into the local optimum in that they are not good at making use of global context in the given region. In addition, when it comes to the target's boundaries, the loss of semantic information through correlation eventually results in imprecise predictions. These mentioned models promote the performance of siamese trackers by using trending structures [3–6] or adding an additional online updater [7–9]. However, due to the inherent defect of the siamese network, which applies correlation as the representation of similarity between the given target and its template, these trackers are not able to achieve a high-level result in most single-object tracking (SOT) benchmarks [10–12].

This linear operation becomes an obstacle when sharp deformation of the object or severe interference from the complex background exists.

To avoid these problems and in pursuit of better speed and accuracy, Transformer [13] and transformer-like structures, which have been proven very efficient in computer vision tasks, are a good choice. Thus, some transplants from picture classification to object tracking were made. The transformer structure overcomes the bottleneck of traditional siamese-based methods, leading to high speed and high accuracy in tracking.

Against the background interference and the object deformation, as Figure 1 shows, we introduce a highly specialized feature fusion network based on Transformer. Transformer-based networks improve the tendency toward local optimum and loss of semantic information necessary for accurate boundary prediction brought about by correlation. This new mechanism improves the quality of similarity representation in many extreme circumstances. Our proposed feature fusion network is based on total attention and applies an unsymmetrical structure that utilizes two different branches in the feature fusion process. For better representation of similarity instead of correlation, our network consists of a direct attention strengthening module and a cross region fusion module in each layer. Our new fusion mechanism not only actively integrates the features of the region of interest and the template but also effectively avoids the mutual interference between them while computing attention. In addition, in order to utilize information from different dimensions, we design a unique combination of the two-layer output of the backbone extractor and decoupling prediction heads in order to achieve better classification and regression. In summary, our main contributions are listed below:

1. We propose a transformer-like feature fusion network based on pure utilization of attention mechanisms, combining the template and the search region instead of a cross-correlation operation;
2. We develop two highly specialized attention modules: a direct attention strengthening module based on self-attention and a cross-region fusion module with cross attention, enabling the tracker to focus on useful information and establish long-term feature associations;
3. We propose decoupling prediction heads for both classification and regression along with the two-layer output mechanism to enhance the results of the previous attention map. Furthermore, we replace the basic GIOU loss with DIOU, which is more suitable for single object tracking.

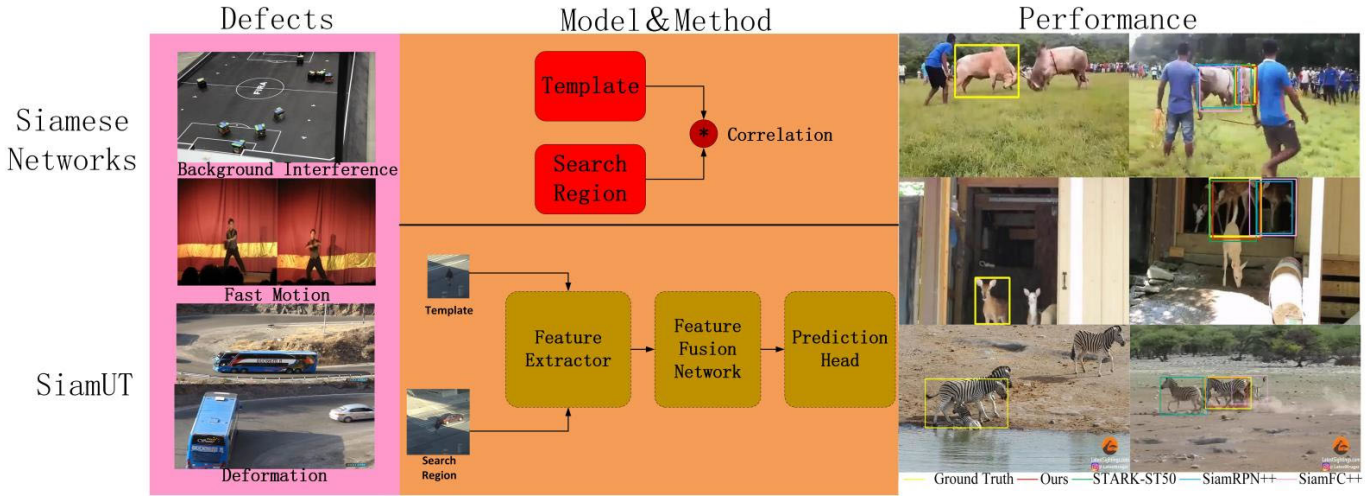

Figure 1. Motivation of our model, SiamUT.

2. Related Works

In this section, we briefly introduce recent siamese trackers since the siamese structure has been found to be especially efficient in visual tracking tasks. In addition, we also present Transformer and attention-based networks that are relevant to our work.

2.1. Siamese Networks Based on Cross-Correlation Operation

In recent years, siamese networks have been a hotspot in visual object tracking [14–18]. Trackers based on the siamese network achieved a good balance between tracking accuracy and efficiency [19–22]. These siamese trackers consider the task a cross-correlation problem between the given search region and the template. Mainstream siamese tracking architectures contain a backbone feature extractor and a correlation-based network to compute similarity between the target and the search region.

SiamFC [14], the pioneer of siamese trackers, first extracts deep features from both the template and the search region with the same trained CNN backbone, then calculates the cross-correlation between the two feature maps to compute the matching scores for target localization. SiamRPN [16], adding the RPN structure [23] widely in order to directly obtain the regression of the bounding box as well as conducting depth-wise correlation. SiamRPN++ [3] proposes a layer-wise feature aggregation structure for the cross-correlation operation and a depth-wise separable correlation structure to produce multiple similarity maps. In addition to these mentioned networks, other popular trackers, including ATOM [24] and DiMP [25], are also highly dependent on the cross-correlation operation. The methodology of these trials is to deepen the feature extractor for better feature maps with the same similarity core—cross-correlation.

However, the cross-correlation itself has two overlooked drawbacks. One defect of this mechanism is that networks using cross-correlation tend to fall into local optimums because of the inability to fully utilize the global context in the designated region from one frame. The other defect is that the cross-correlation operation itself is bound to lose semantic information to some degree. When it comes to the boundaries of the object, this inevitable loss of high-level information is the reason for imprecision. To avoid these mentioned issues, our work introduces a Transformer-like feature fusion network using attention mechanisms instead of cross-correlation.

2.2. Transformer and Transformer-like Networks

Transformer is first applied in machine translation in the field of NLP. It has replaced RNN in many tasks, like language and speech processing [26–28]. Briefly, Transformer has encoders and decoders, both based on attention. These modules transform one sequence into another and generate output tokens one by one. In computer vision, the parallel encoders and decoders based on attention mechanisms help Transformer function as well as in NLP.

DETR [29], a Transformer encoder-decoder architecture that forces unique predictions via a bipartite matching procedure. On the challenging COCO [30] dataset, DETR performs much better in comparison to the Faster R-CNN baseline [23]. Motivated by the success of DETR in detection, Transt [31] attempts to bring Transformer into the tracking field, considering the similarity between detection and tracking. Transt does not simply copy the encoder-decoder architecture. To apply Transformer to tracking, Transt designs a symmetrical module that is a combination of self-attention and cross attention for both the template and the search region to be input. Other efforts have also been made to introduce the attention mechanism into tracking tasks, such as SiamAttn [9], which also combines the self-attention branch and the cross-attention branch but still applies depth-wise correlation, and SparseTT [32], which relieves the problem that self-attention lacks focus on the relevant information by modifying the network with a sparse attention mechanism.

These methods replace correlation with transformer-like structure in feature fusion networks. However, there is still one deficiency that has not been solved by the introduction of Transformer yet. The problem lies in that during the symmetrical self-attention and cross-attention operations for both the template and the search region, the feature map of each layer mingles relevant and irrelevant information together equally, causing the attention mechanism to become confused about what to focus on. That is to say, at certain locations in the transformer-like architecture, the two branches have to be different from each other. Inspired by this concept, we preserve the vital idea and aspects of Transformer and design a new unsymmetrical feature fusion network based on attention mechanisms.

3. Model

In Figure 2, we present the SiamUT in the form of a flow chart. Our framework consists of three important components: a feature extractor using the ResNet50 backbone [33], a feature fusion network, and a decoupling prediction head network for localizing the target. In the beginning section, we introduce the details of the feature extractor. Afterwards, we demonstrate how the feature fusion network, based entirely on attention, and the two significant modules work to process the features. At the end of this chapter, we have some discussion about the advanced decoupling head network.

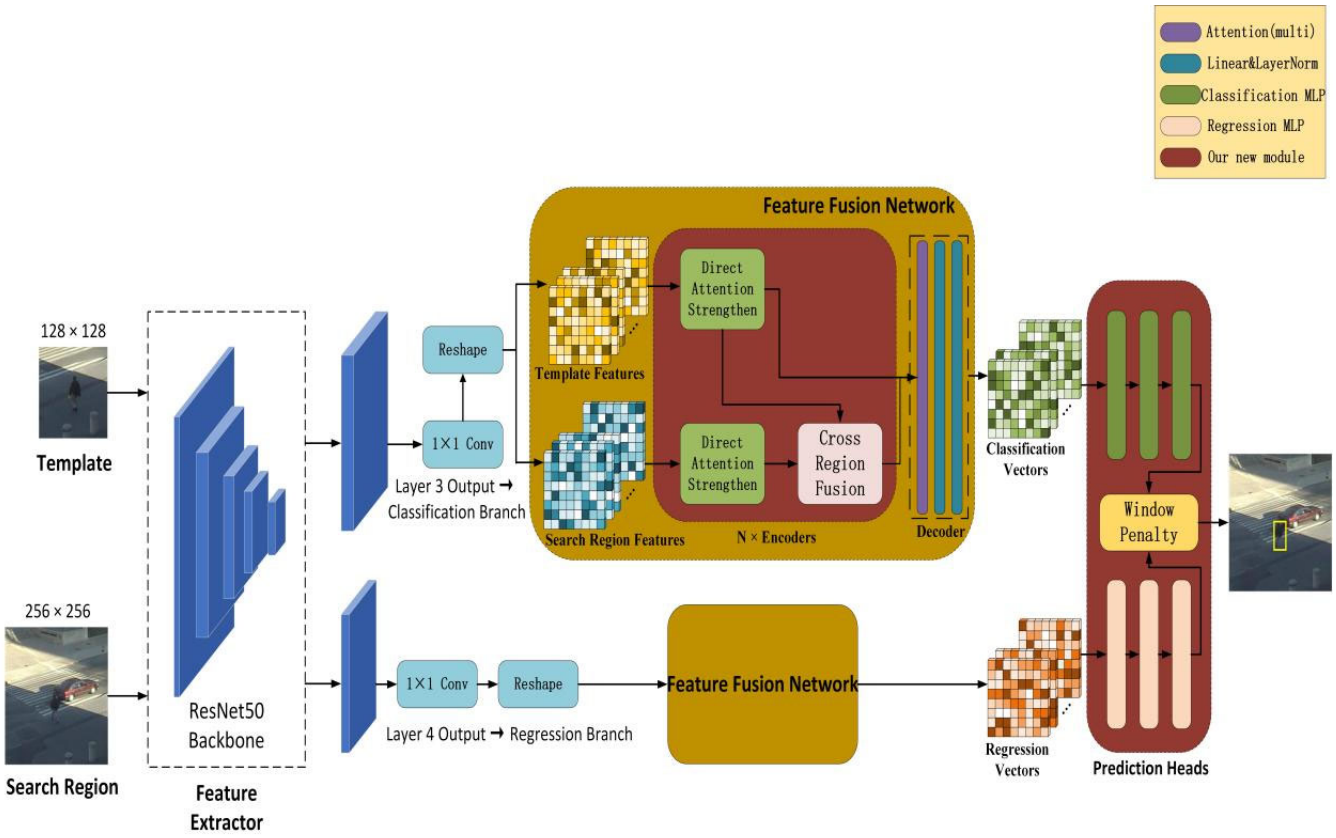

Figure 2. Architecture of our SiamUT framework. Our model consists of three basic components: a feature extractor (ResNet50 Backbone), a feature fusion network (both layer 3 and layer 4 output), and prediction heads.

3.1. Feature Extractor

The proposed SiamUT network operates like some siamese-based trackers. Its feature extractor first crops a pair of image patches as the original input of the backbone, i.e., the template image $Z \in \mathbb{R}^{h_z \times w_z \times 3}$ and the search region image $X \in \mathbb{R}^{h_x \times w_x \times 3}$ as well. In order to obtain both appearance information from the target object and some necessary background information from its surroundings, the template image is expanded by twice

the side length from the center of the given object in the first frame of its video sequence. For the search region, it is required to cover as much of the target-accessible area as possible. So that the search region image is expanded by four times the side length from the center of the coordinate object in the previous frame.

In order to obtain both appearance information from the target object and some necessary background information from its surroundings, the template image is expanded by twice the side length from the center of the given object in the first frame of its video sequence. For the search region, it is required to cover as much of the target-accessible area as possible. So that the search region image is expanded by four times the side length from the center of the coordinate object in the previous frame. After that, feature maps F_z and F_x of the template and the search region are generated by the backbone. It is noted that $F_Z \in \mathbb{R}^{h_{zl} \times w_{zl} \times c_l}$ and $F_x \in \mathbb{R}^{h_{xl} \times w_{xl} \times c_l}$, h_{zl}, h_{xl}, w_{zl}, w_{xl}, and c_l are constants in coordinate with their layers in the backbone.

There are lots of optional backbones for feature extractors such as AlexNet [34], ResNet, DenseNet [35], and huge structures using attention. As described in [36], among these backbones, ResNet is not that complicated and rich in both semantic and localization information due to its multi-layer output, which is able to assist subsequent networks in classification and regression. Thus, we chose ResNet50 for the feature extractor. Especially, layers 3 and 4 are both selected as the output layers. The deeper layer, rich in semantic information, provides regression vectors only, while the other one provides classification vectors only.

3.2. Feature Fusion Network

Different from the traditional symmetrical transformer-like structure [31], we propose an unsymmetrical feature fusion network with direct attention strengthen modules (DAS) and cross-region fusion modules (CRF).

Before entering the feature fusion network, the channel dimensions of F_z and F_x need to be reduced. Here we use a common 1×1 convolution to accomplish that. In the two low-dimensional feature maps $F^{'}{}_z$ and $F^{'}{}_x$, $F_z^{'} \in \mathbb{R}^{h_{zl} \times w_{zl} \times d}$, $F_x^{'} \in \mathbb{R}^{h_{xl} \times w_{xl} \times d}$, d is the default compact number of dimensions. The succeeding transformer-like feature fusion network takes vectors as input, so the two low-dimensional feature maps also have to become feature vectors. $F^{'}{}_z$ and $F^{'}{}_x$ are flattened in spatial dimension, obtaining the required feature vectors f_z and f_x, $f_z \in \mathbb{R}^{h_{zl} \times w_{zl} \times d}$, $f_x \in \mathbb{R}^{h_{xl} \times w_{xl} \times d}$. Both f_z and f_x can be considered piles of vectors of the same length d.

As shown in Figure 2, the two outputs of the feature extractor (taken layer 4 as an example) are sent into the template branch and the search branch as the input correspondingly. In the template branch, the DAS module with multi-head self-attention focuses on the template object itself and its surroundings. The output feature vectors containing information about the template and its background in this branch are also the input of the CRF module in the search branch and the decoder. In the search branch, the incoming feature vectors are also processed by the DAS module at first. Then the processed search region feature vectors, together with the processed template feature vectors, are taken into the CRF module. The CRF module receives feature vectors from the two branches at the same time and fuses these different features with its built-in multi-head cross attention. The two DAS modules and the CRF module function as a feature fusion layer, as well as the encoder in the origin transformer. After that, the fused feature vectors of the CRF module and the template vectors of the DAS module are sent into the decoder. The decoder uses multi-head cross attention to process the two inputs from the two branches and eventually obtains the regression vectors or classification vectors.

Multi-head attention is a significant mechanism in our proposed feature fusion network. Since the scale dot-product attention function is the basic attention mechanism, the details of it are not repeated here. [13] developed the basic attention into a multi-head version. For a better understanding of our network, we present the detailed descriptions of multi-head attention according to [13].

Multi-head attention is able to take various attention distributions into consideration and focus on different regions of the input. Multi-head attention is defined in Equations (1)–(3), where Q represents queries, K represents keys, V represents values, weight matrices $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_k}$ and $W^O \in \mathbb{R}^{d_m \times d_v \times n_h}$; n_h, d_m, $dk = d_v = d_m/n_h$ are default values.

$$Multi - head(Q, K, V) = Concat\left(H_1, \ldots, H_{n_h}\right) W^O \tag{1}$$

$$H_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{2}$$

$$\text{Attention}(Q, K, V) = Softmax\left(\frac{QK}{\sqrt{d_k}}\right) V \tag{3}$$

The structure of DAS is shown in Figure 3. The main function of the DAS module is to integrate information from various parts of the feature map. Combining the residual [33] formation and multi-head attention, DAS is able to strengthen the given feature map according to the default number of channels. Due to the lack of ability to distinguish positional information, DAS inputs should be positional encoded first. Our method for generating positional encoding is a sine function. The following Equation (4) is the summary of the DSA mechanism, where $P_Y \in \mathbb{R}^{d \times N_Y}$ is the outcome of the sine positional encoding module and $Y_{DAS} \in \mathbb{R}^{d \times N_Y}$ is the output of the whole DAS module.

$$Y_{DAS} = Y + Multi - head(P_Y, P_Y, Y) \tag{4}$$

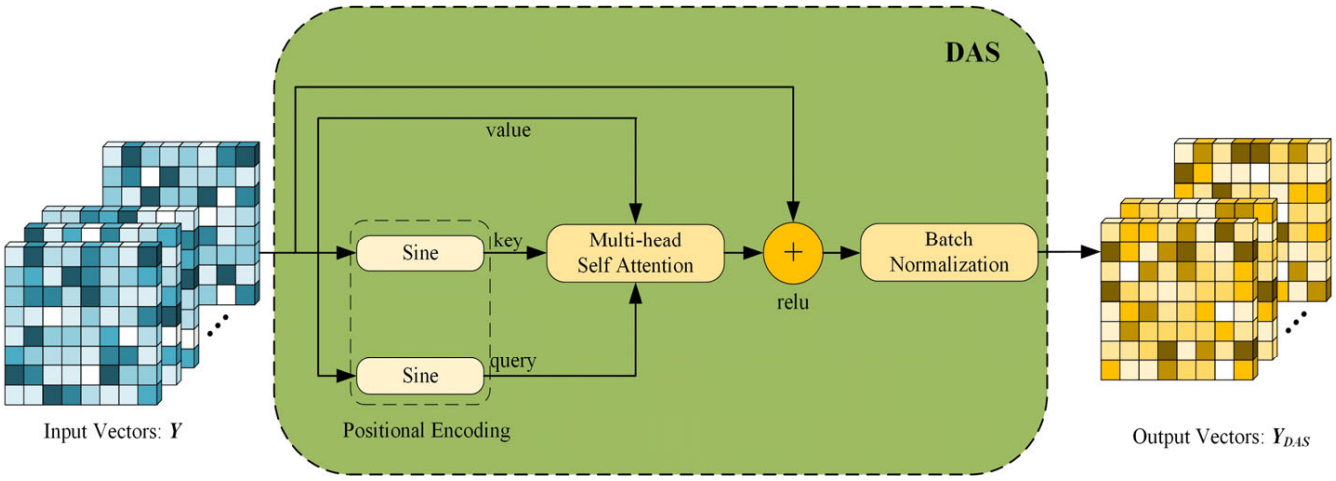

Figure 3. Direct Attention Strengthen module (DAS) is based on multi-head self-attention and residual structure.

The structure of CRF is shown in Figure 4. The main function of the CRF module is to fuse feature vectors from both branches. The CRF module takes the output of its previous DAS module in the search branch and the output of the DAS module in the template branch as its input. Because of the similar attention structure, inputs to CRF also need to be sine-positional encoded before calculating. As described in [37], it is helpful in promoting the fitting ability of the model to add an additional fully connected feed-forward network module (FFN) after the attention part. Considering the balance between the benefit of FFN and the increase in parameters by adding FFN, we only plant a FFN module in CRF. The

FFN module in CRF is a 3-layer linear transformation with rectified linear units (ReLU). The following Equations (5)–(7) are a summary of the CRF mechanism.

$$\widetilde{Y_{CRF}} = Y_S + Multi - head\left(P_{Y_s}, P_{Y_T}, Y_T\right) \tag{5}$$

$$Y_{CRF} = \widetilde{Y_{CRF}} + FFN\left(\widetilde{Y_{CRF}}\right) \tag{6}$$

$$FFN(x) = max(0, max(0, xW_1 + b_1)W_2 + b_2)W_3 + b_3 \tag{7}$$

where $Y_S \in \mathbb{R}^{d \times N_s}$ is the output of the previous DAS module in the search branch, $Y_T \in \mathbb{R}^{d \times N_T}$ is the output of the DAS module in the template branch, $P_{Y_S} \in \mathbb{R}^{d \times N_s}$ is the outcome of sine positional encoding of Y_S in the search branch, $P_{Y_T} \in \mathbb{R}^{d \times N_T}$ is the outcome of sine positional encoding of Y_T in the template branch, $\widetilde{Y_{CRF}}$ is the outcome of the multi-head attention stage, Y_{CRF} is the output of the CRF module. $W_{1,2,3}$ and $b_{1,2,3}$ are weight matrices and bias matrices in different layers of the FFN module.

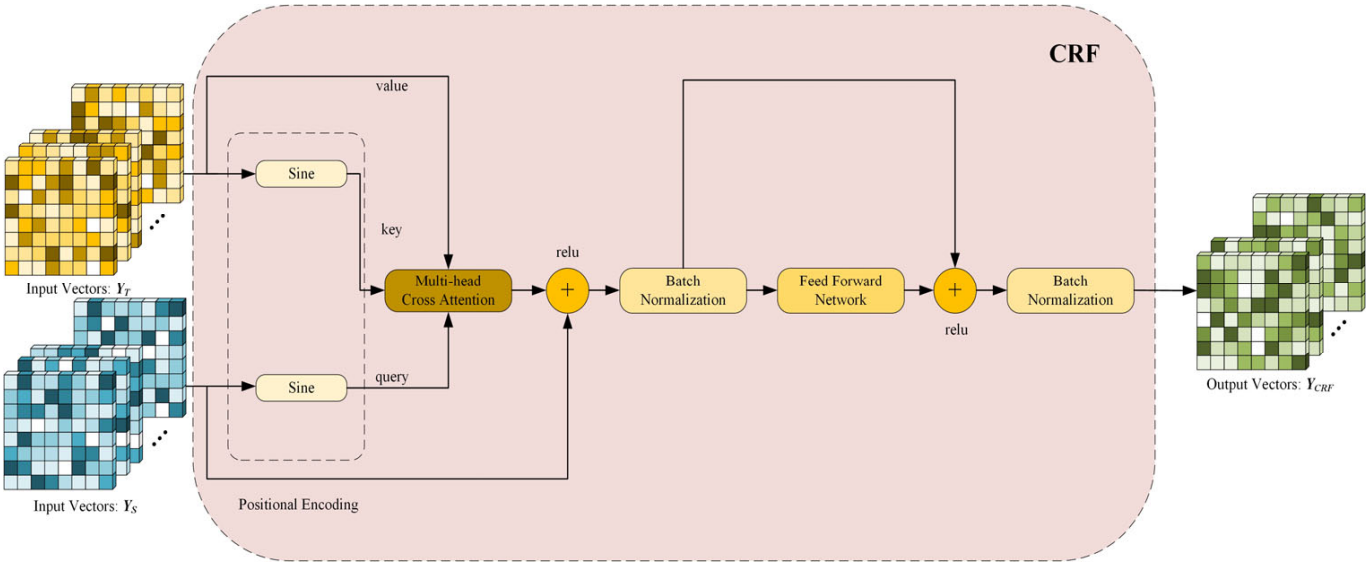

Figure 4. Cross Region Fusion module (CRF) based on multi-head cross-attention and a fully forward network with residual structure.

CRF integrates the features of the template and the search region. Two DAS modules and one CRF module constitute a feature fusion layer where features of the given object and the background of the search area are integrated. More feature fusion layers improve the result of fusing feature maps and slow down the speed of operation at the same time. In our experiment, we used two feature fusion layers. Also, the feature fusion layers act as the encoder in the transformer structure. The following decoder takes the output of previous feature fusion layers as input. Finally, the output of the decoder can be used by the decoupling prediction head.

3.3. Prediction Heads and Training Loss

Our prediction network consists of two decoupling prediction heads in two different branches: one is the classification branch, and the other is the regression branch. In convolutional networks, deeper layers have more semantic information, while shallow layers have more mechanical information like location and color. Thus, we allocate different output layers to different branches. As shown in Figure 2, the regression branch receives

the layer 4 output of the feature extractor, while the classification branch receives the layer 3 output of the feature extractor.

Our prediction network completely abandons the anchor points, or anchor boxes, based on prior knowledge. The decoupling prediction heads of two branches generate predict logits, which contain foreground/background classification results and normalized coordinates on the basis of the original size of the search region from the feature vectors of the feature fusion network. The tracker can compute the ultimate bounding box by predicting logits directly. That is why our 2-layer decoupling prediction heads could enhance the precision of the whole tracking network. Each of them is a multi-layer perceptron (MLP). In our implementation, the number of MLP layers is three.

Receiving feature vectors from the previous feature fusion layer, each prediction head generates binary prediction logits containing classification or regression results according to which branch the head is in. For classification, positive samples are the predictions of feature vectors relevant to the pixels in the given ground-truth bounding box, while negative samples are the rest. In the regression branch, only positive samples have an effect on the total regression loss. In the classification branch, the total classification loss is relevant to all the positive and negative samples. We define our classification loss with the standard binary cross entropy loss in Equation (8) as follows:

$$Loss_{cls} = -\sum_{i}(1 - z_i)log(1 - p_i) + z_i log(p_i) \tag{8}$$

where z_i is the i-th sample's ground truth label, $z_i = 1$ when it is in the foreground, and $z_i = 0$ when it is in the background. P_i is the probability belonging to the foreground of the prediction generated by the prediction head.

Also, for bounding box regression, we define our regression loss with a linear combination of the L1-norm loss [38] and the Distance-IoU (DIoU) loss [39] in Equation (9) as follows:

$$Loss_{\text{reg}} = \sum_{i} F_{\{z_i \in P\}} \left[\rho_d L_{DIoU}\left(z_i, \hat{z}\right) + \rho_1 L_1\left(z_i, \hat{z}\right) \right] \tag{9}$$

where F is the indicator function, P is the set of positive samples, z_i denotes the i-th predicted bounding box, ρ_d is the weight of DIou loss in the total regression loss, and ρ_1 is the weight of L1-norm loss in the total regression loss.

4. Experiments

4.1. Implementations

In the offline training stage, we train the SiamUT model on LaSOT [40], GOT-10k [41], and TrackingNet [42] datasets. For these video datasets, we split and sampled the videos for training. First, the dataset is selected at random. Next, from that chosen dataset, we pick a sequence. Then the base frame is sampled from the sequence. After that, the set of template frames and search frames are sampled from the sequence in a default range, respectively. It should be noted that only the frames in which the given target is at least visible to some extent could be sampled. Actually, the training is based on these sampled data splits. For LaSOT, we divide it into three parts: training, validation, and testing. The proportion is 3:1:1. For TrackingNet, in which the testing set is given, we only need to divide the rest of it into 2 parts by 5:1. Especially, 1000 videos are removed from the GOT-10k dataset. This is a fair comparison, according to [43]. As well as TrackingNet, we still use the given testing set, and the proportion between training and validation is 4:1.

We follow the common processing rules [14] for siamese-based tracking. The search region is cropped from the current frame, and its center is the predicted position of the last frame. For the search region patch, the size is 256 $\times$ 256, while the size of the template patch is 128 $\times$ 128. The parameters of our ResNet-50 backbone are initialized with the one pretrained on ImageNet [44], and the parameters of other parts of our model are initialized with Xavier [45]. The optimizer in training is AdamW [46], in which the learning rate of the backbone is 2×10^{-5}, the learning rate of other parts is 2×10^{-4}, and the

weight decay is 1×10^{-4}. We train the network on one GTX 3080 GPU for 800 epochs with 1000 samples per epoch. The learning rate drops by a factor of 10 after 400 epochs. In respect of hyperparameters in our model, length d = 256 after output layer 3. In both the DAS module and the CRF module, $n_h = 8$, d_m = d, $d_k = d_v = 32$. In the loss function, weight factors $\rho_d = 4.5$ and $\rho_1 = 2.3$.

In the online tracking stage, we use the window penalty to post-process the output bounding boxes of the prediction head. In the penalty function, we apply a 32×32 Hanning window weighted by 0.49 to the confidence scores of those boxes. According to the penalty, feature points far from the target in the previous frame become punished in their confidence scores, so those unlikely bounding boxes could be excluded. The ultimate tracking result is the bounding box with the highest confidence score.

4.2. Evaluation

In this subsection, we conduct experiments and compare the performance of SiamUT with other state-of-the-art trackers on three benchmarks.

LaSOT [40] is a dataset consisting of 1400 video clips with more than 3.5 million frames in total. This benchmark is widely applied in measuring trackers long term capability. The average video length in LaSOT is 2500 frames. Each of the video sequences comprises up to 15 challenging attributes. A one-pass evaluation protocol (OPE) is used to measure the normalized precision PNorm and the area under the curve (AUC) of the success plot. Table 1 shows the comparison between our model, SiamUT, and other SoTA trackers on the LaSOT benchmark.

GOT-10k [41] is also a large-scale dataset. It contains ten thousand training video sequences and 180 clips for testing. In this benchmark, the average overlap (AO) and the success rate (SR) at overlap thresholds of 0.5 and 0.75 are adopted. Our model is retrained on GOT-10k train splits following its unique evaluation protocol. Table 2 shows the comparison between SiamUT and other SoTA trackers on the GOT-10k benchmark.

TrackingNet [42] is another large-scale dataset with about 30,000 training video sequences and 511 testing splits. All the trackers are evaluated under the same two indicators as LaSOT on the test splits through its evaluation server. Table 3 shows the comparison between our model and other SoTA trackers on the TrackingNet benchmark.

Table 1. Comparison with state-of-the-art trackers on LaSOT (The best results are shown in bold).

Trackers	**AUC (%)**	P_{Norm} **(%)**
SiamRPN++ [3]	49.6	56.9
SiamFC++ [4]	54.4	62.3
PACNet [47]	55.3	62.8
Ocean [5]	56.0	65.1
DiMP50 [48]	56.9	64.3
Transt [31]	64.7	73.8
SiamR-CNN [49]	64.8	72.2
STARK-ST50 [43]	66.1	**76.3**
Ours	**66.5**	75.5

Table 2. Comparison with state-of-the-art trackers on GOT-10k (the best results are shown in bold).

Trackers	AO (%)	$SR_{0.5}$ (%)	$SR_{0.75}$ (%)
SiamRPN++ [3]	51.7	61.6	32.5
SiamFC++ [4]	59.5	69.5	47.9
SiamCAR [50]	56.9	67.0	41.5
Ocean [5]	61.1	72.1	47.3
DiMP50 [48]	63.4	73.8	54.3
Transt [31]	66.2	75.5	58.7
SiamR-CNN [49]	64.9	72.8	59.7
STARK-ST50 [43]	**68.0**	**77.7**	**62.3**
Ours	67.5	76.5	60.3

Table 3. Comparison with state-of-the-art trackers on TrackingNet (the best results are shown in bold).

Trackers	AUC (%)	P_{Norm} (%)
SiamRPN++ [3]	73.3	80.0
SiamFC++ [4]	75.4	80.0
SiamAttn [9]	75.2	81.7
CGACD [8]	71.1	81.0
DiMP50 [48]	74.0	80.1
Transt [31]	81.4	86.7
SiamR-CNN [49]	81.2	85.4
STARK-ST50 [43]	81.3	86.1
Ours	**82.4**	**87.0**

It should noted that our model SiamUT achieves comparable performance with all other state-of-the-art trackers on the LaSOT and TrackingNet benchmarks, while on the GOT-10k benchmark, STARK-ST50 obtains the best score. In our view, this result is probably due to the different evaluation protocols mentioned between GOT-10k and the other two benchmarks. Figures 5–7 show the tracking results of our model and other state-of-the-art trackers in three challenging conditions, including sharp deformation of the object, interference from the background, and occlusion of the object. These video clips are selected from the mentioned datasets at random. Our SiamUT merely has no error with the ground truth bounding box compared with other trackers. Table 4 shows the AUC scores of our model as well as those of some other state-of-the-art trackers in several challenging scenarios on the LaSOT dateset. This experiment proves that our tracker is more precise when dealing with complicated tasks, including partial/full occlusion, background clutter, and deformation. In general, our tracker, SiamUT, achieves state-of-the-art performance in both benchmarks and shows superior results by a large margin when compared to other recently proposed models.

Table 4. AUC score comparison of challenging attributes on LaSOT (The best results are shown in bold).

Trackers	SiamRPN++ [3]	STARK-ST50 [43]	ATOM [24]	DiMP [48]	Ocean [5]	Ours
Partial Occlusion	46.5	58.2	47.4	51.5	50.9	**62.1**
Full Occlusion	37.4	52.4	41.8	51.0	42.3	**55.5**
Deformation	53.2	62.8	52.2	57.1	62.5	**66.9**
Background Clutter	44.9	55.0	45.0	48.8	54.3	**56.4**

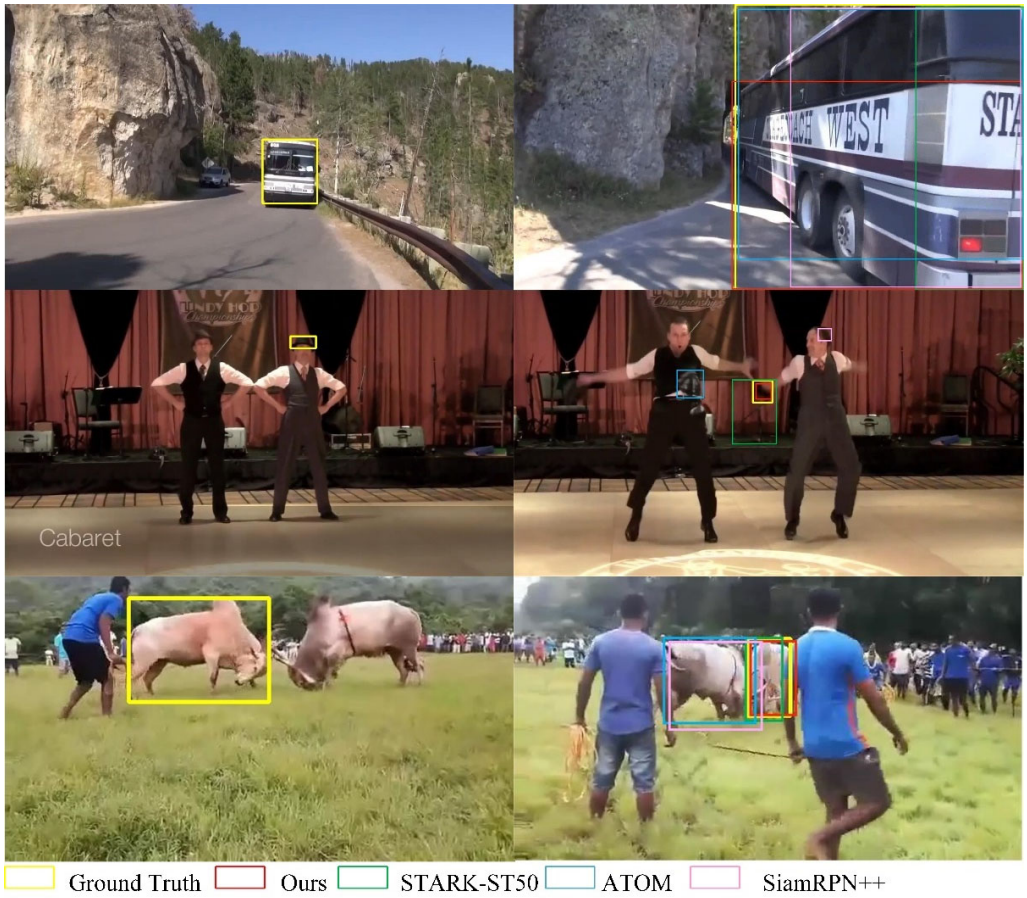

Figure 5. Tracking results of SiamUT and three other state-of-the-art trackers under object deformation.

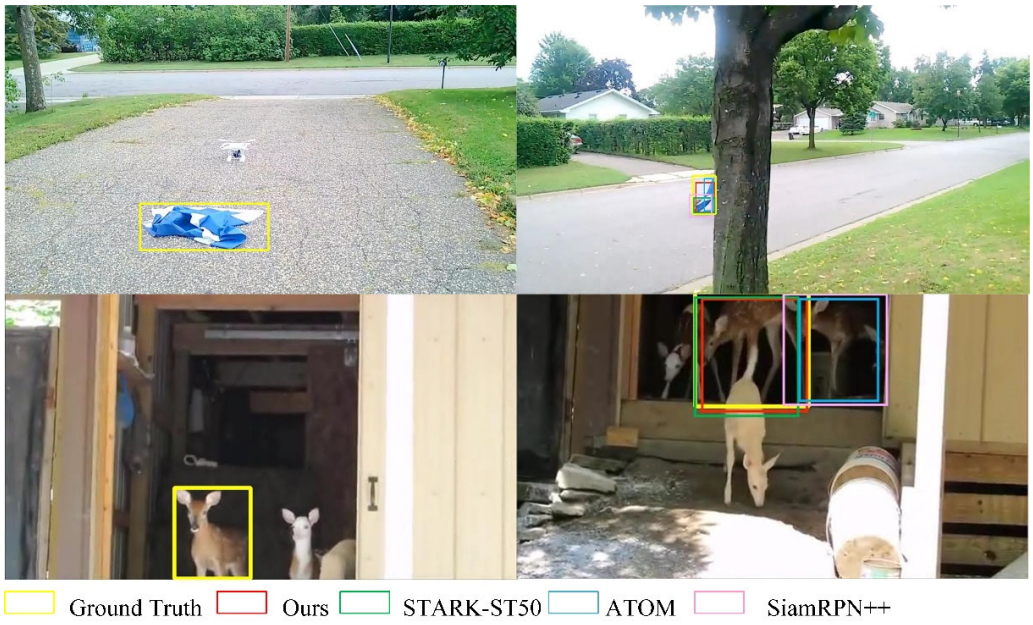

Figure 6. Tracking results of SiamUT and three other state-of-the-art trackers under object occlusion.

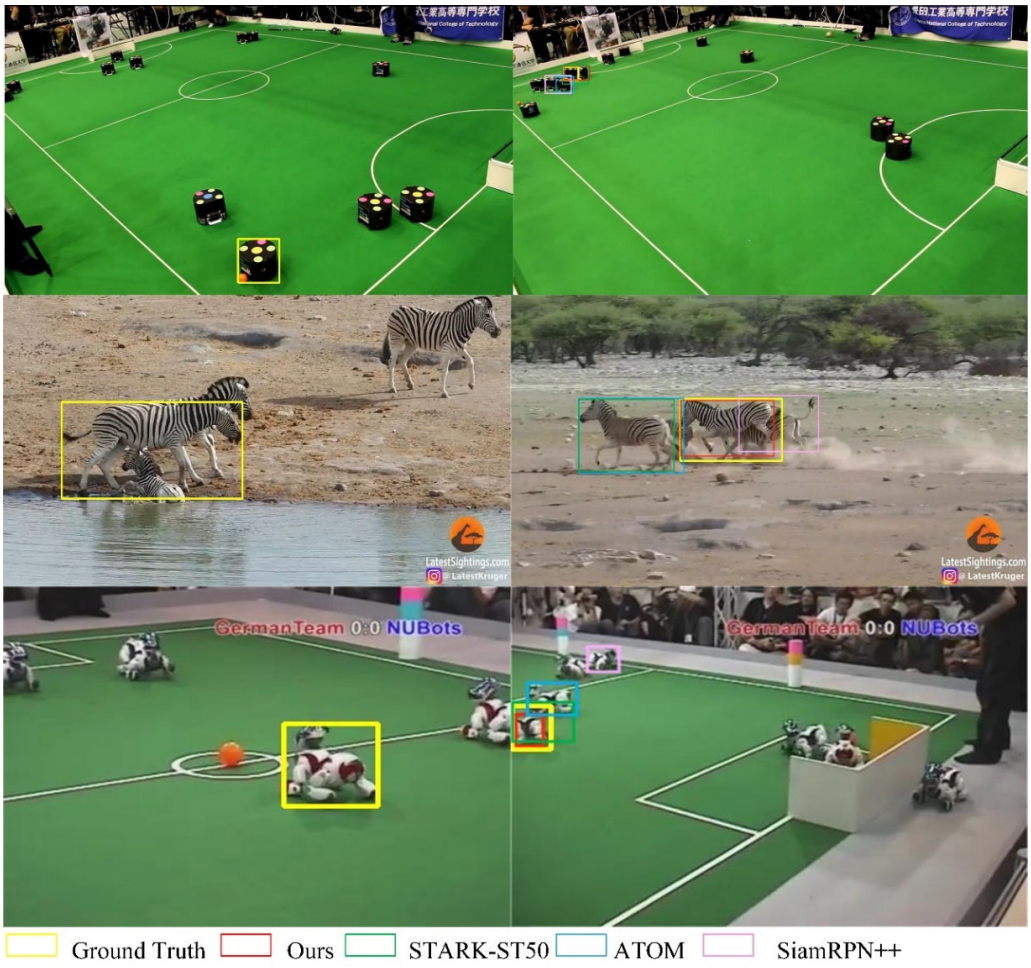

Figure 7. Tracking results of SiamUT and three other state-of-the-art trackers under background interference.

4.3. Ablation

In order to indicate the superiority of our designed feature fusion network using DAS and CRF modules, we compare our tracker with the transformer using the original structure in tasks of computer vision. Our feature fusion network with built-in multi-head self-attention (in DAS) and built-in multi-head cross region attention (in CRF) is quite different from the original structure. To build the original transformer model, the encoder obtains the template features as input and the decoder obtains the search region features as input. For that reason, the output size of the encoder should be correlated with the input size of the decoder. Results of this comparison are shown in Table 5, where the transformer denotes the basic and original transformer structure while DAS and CRF denote our modified modules.

The next is the comparison between our two-layer output and the ResNet50 backbone. The original structure only uses the output of layer 3. Both the template features and search regions are generated from layer 3, and the succeeding feature fusion network and the prediction head operate on this base. That is to say, the unified prediction head only utilizes the information from the layer 3 of the backbone for both classification and regression. Our two-layer output and decoupled prediction heads improve information efficiency. The features of the two branches from layer 3 are for the classification prediction head. Template and search region features from the deeper layer 4 containing more semantic information are for the regression prediction head. Results of this comparison are in Table 5, where the two-layer output and decoupling prediction heads denote the special pre-

processing operation before the feature fusion network and our modified structure in the prediction head.

Table 5. Ablation results on LaSOT and TrackingNet (the best results are shown in bold).

Method						**LaSOT**		**TrackingNet**	
Transformer	**DAS and CRF**	**Two-Layer Output**	**Decoupling Prediction Heads**	**GIoU**	**DIoU**	**AUC (%)**	P_{Norm} **(%)**	**AUC (%)**	P_{Norm} **(%)**
√				√		64.2	73.7	81.1	86.8
	√			√		65.1	74.0	81.6	86.6
	√	√		√		63.0	71.3	80.7	85.1
	√		√	√		63.9	73.1	80.9	86.2
	√	√	√	√		66.3	75.5	82.3	86.8
	√	√	√		√	**66.5**	**75.5**	**82.4**	**87.0**

The last comparison is between the Distance-IoU loss and the generalized IoU loss (GIoU) [51]. Bounding box regression is one of the fundamental components of many 2D/3D computer vision tasks. Generalized IoU loss is widely proposed to the benefit of the IoU metric. In the algorithm of the generalized IoU loss function, GIoU is attained by subtracting the ratio from the IoU value, keeping the major properties of IoU while rectifying its weakness. Therefore, GIoU is a proper substitute for IoU in our single-object tracking task. The DIoU loss in incorporating the normalized distance between the given target bounding box and the predicted one. By directly minimizing the normalized distance between the central points of the two bounding boxes and providing moving directions for bounding boxes when non-overlapping, DIoU loss has several merits over the original IoU. Summarizing geometric factors including overlap area, central point distance, and aspect ratio, DIoU overcomes common problems like slow convergence and inaccurate regression and converges much faster in training. The result of the comparison between adopting the GIoU loss and the DIoU loss is shown in Table 5.

According to Table 5, the insertion of DAS and CRF modules increases the area under the curve of both LaSOT and TrackingNet benchmarks by 0.7% on average, with almost no negative effect on normalized precision. This proves that our specialized modules for single object tracking are better than the original transformer structure to some extent. In the next few lines, statistics show that inserting the two-layer output separately leads to a discriminating decrease on both AUC and normalized precision, while the decrease of applying the decoupling prediction heads is much smaller. Compared to the results of using the two-layer output together with the decoupling prediction heads, we found that the separation of these two modules deconstructs the consistency within the model, which leads to poor performance on both benchmarks. That is to say, the two-layer output module needs to be paired with its proceeding module, the decoupling prediction heads. The unification of this two modules increases about 2% on both indicators on LaSOT benchmark, which is a significant improvement. The last two lines compare the Distance-IoU with the generalized IoU. It is obvious that the updated DIoU is more suitable for single-object tracking tasks. The utilization of DIoU brings a 0.2% improvement on both benchmarks.

5. Conclusions

In this work, we propose an unsymmetrical siamese structure with a Transformer-like feature fusion network based on attention mechanisms. We also modify the backbone feature extractor and the prediction network in coordination with the feature fusion stage.

Extensive experiments prove the effectiveness of our design. Compared with other state-of-the-art trackers, our model is more robust and accurate when handling various challenging tracking tasks. Especially when dealing with severe background interference, including similar objects along with occlusion and deformation, our SiamUT has been

proven to have certain advantages in accuracy. Our design could be applied to monitor systems both in cities and in the wild. Tracking a vehicle, person, or animal from its kind via a monitor is an appropriate application of our design.

Furthermore, we are going to improve our method in two aspects. One is adapting our model to multiple object tracking (MOT) for practical tracking tasks. This could make our method a unified tracking algorithm. The other is applying simulated data. As proved in [52,53], simulated data could be appropriate for highly specialized tracking objects. The application of simulated data including pedestrians or vehicles may be helpful given the scarcity of such objects in the datasets mentioned in Section 4.

Finally, we hope our work can be helpful to future research and industrial applications.

Author Contributions: Methodology, L.Y. and M.X.; software, L.Y. and D.C.; data curation, L.Y., Z.S., and E.C.; writing—original draft preparation, L.Y.; writing—review and editing, L.Y., G.Y., and H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Special Fund for Key Program of Science and Technology of Yunnan Province, China, grant number 202202AD080004.

Data Availability Statement: Datasets in this study are all public datasets.

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

References

1. Wu, Y.; Lim, J.; Yang, M.-H. Object tracking benchmark. *Proc. IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
2. Zhang, X.; Chen, J.; Yuan, J.; Chen, Q.; Wang, J.; Wang, X.; Han, S.; Chen, X.; Pi, J.; Yao, K.; et al. Cae v2: Context autoencoder with clip target. *arXiv* **2022**, arXiv:2211.097993. [CrossRef]
3. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. *arXiv* **2019**, arXiv:1812.11703. [CrossRef]
4. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. *arXiv* **2020**, arXiv:1911.06188. [CrossRef]
5. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-Aware Anchor-Free Tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020. [CrossRef]
6. Zheng, L.; Tang, M.; Chen, Y.; Wang, J.; Lu, H. Learning Feature Embeddings for Discriminant Model Based Tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020. [CrossRef]
7. Choi, J.; Kwon, J.; Lee, K.M. Deep meta learning for real-time target-aware visual tracking. *arXiv* **2019**, arXiv:1712.09153. [CrossRef]
8. Du, F.; Liu, P.; Zhao, W.; Tang, X. Correlation-Guided Attention for Corner Detection Based Visual Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
9. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. *arXiv* **2020**, arXiv:2004.06711. [CrossRef]
10. Bergmann, P.; Meinhardt, T.; Leal-Taixe, L. Tracking without Bells and Whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019. [CrossRef]
11. Sadeghian, A.; Alahi, A.; Savarese, S. Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. [CrossRef]
12. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
14. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 15–16 October 2016. [CrossRef]
15. Held, D.; Thrun, S.; Savarese, S. Learning to Track at 100 fps with Deep Regression Networks. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016. [CrossRef]
16. Li, B.; Yan, J.; Wu, W.; Zhu, 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.
17. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese Instance Search for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1420–1429.

18. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813. [CrossRef]
19. Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; Hu, W. Dcfnet: Discriminant correlation fifilters network for visual tracking. *arXiv* **2017**, arXiv:1704.04057. [CrossRef]
20. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning Attentions: Residual Attentional Siamese Network for High Performance Online Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4854–4863.
21. Wang, Q.; Zhang, M.; Xing, J.; Gao, J.; Hu, W.; Maybank, S.J. Do Not Lose the Details: Reinforced Representation Learning for High Performance Visual Tracking. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm Sweden, 13–19 July 2018; Available online: https://eprints.bbk.ac.uk (accessed on 16 July 2023).
22. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware siamese networks for visual object tracking. *arXiv* **2018**, arXiv:1808.06048. [CrossRef]
23. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R–CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
24. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate tracking by overlap maximization. *arXiv* **2019**, arXiv:1811.07628. [CrossRef]
25. Bhat, G.; Danelljan, M.; Gool, L.V.; Timofte, R. Learning discriminative model prediction for tracking. *arXiv* **2019**, arXiv:1904.07220. [CrossRef]
26. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2019**, arXiv:1810.04805. [CrossRef]
27. Luscher, C.; Beck, E.; Irie, K.; Kitza, M.; Michel, W.; Zeyer, A.; Schluter, R.; Ney, H. RWTH ASR Systems for LibriSpeech: Hybrid vs attention. *arXiv* **2019**, arXiv:1905.03072. [CrossRef]
28. Synnaeve, G.; Xu, Q.; Kahn, J.; Grave, E.; Likhomanenko, T.; Pratap, V.; Sriram, A.; Liptchinsky, V.; Collobert, R. End-to-end ASR: From supervised to semi-supervised learning with modern architectures. *arXiv* **2019**, arXiv:1911.08460. [CrossRef]
29. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020. [CrossRef]
30. Lin, T.-Y.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014. [CrossRef]
31. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. *arXiv* **2021**, arXiv:2203.13533. [CrossRef]
32. Fu, Z.; Fu, Z.; Liu, Q.; Cai, W.; Wang, Y. SparseTT: Visual Tracking with Sparse Transformers. *arXiv* **2022**, arXiv:2205.03776. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012. [CrossRef]
35. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. *arXiv* **2016**, arXiv:1608.06993. [CrossRef]
36. Lin, L.; Fan, H.; Zhang, Z.; Xu, Y.; Ling, H. SwinTrack: A Simple and Strong Baseline for Transformer Tracking. *arXiv* **2021**, arXiv:2112.00995. [CrossRef]
37. Zhang, Z.; Lin, Y.; Liu, Z.; Li, P.; Sun, M.; Zhou, J. MoEfication: Transformer Feed-forward Layers are Mixtures of Experts. *arXiv* **2022**, arXiv:2110.01786. [CrossRef]
38. Dedieu, A.; Lázaro-Gredilla, M.; George, D. Sample-Efficient L0-L2 Constrained Structure Learning of Sparse Ising Models. *arXiv* **2020**, arXiv:2012.01744. [CrossRef]
39. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020. [CrossRef]
40. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
41. Huang, L.; 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. [CrossRef] [PubMed]
42. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. *arXiv* **2018**, arXiv:1803.10794. [CrossRef]
43. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. *arXiv* **2021**, arXiv:2103.17154. [CrossRef]
44. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]

45. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010.
46. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2018**, arXiv:1711.05101. [CrossRef]
47. Zhang, D.; Zheng, Z.; Jia, R.; Li, M. Visual Tracking via Hierarchical Deep Reinforcement Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2021. [CrossRef]
48. Chen, B.; Wang, D.; Li, P.; Wang, S.; Lu, H. Real-Time 'Actor-Critic' Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
49. Voigtlaender, P.; Luiten, J.; Torr, P.H.S.; Leibe, B. Siam R-CNN: Visual Tracking by Redetection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
50. 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 Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [CrossRef]
51. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
52. Staniszewski, M.; Foszner, P.; Kostorz, K.; Michalczuk, A.; Wereszczyński, K.; Cogiel, M.; Golba, D.; Wojciechowski, K.; Polański, A. Application of Crowd Simulations in the Evaluation of Tracking Algorithms. *Sensors* **2020**, *20*, 4960. [CrossRef] [PubMed]
53. Ciampi, L.; Messina, N.; Falchi, F.; Gennaro, C.; Amato, G. Virtual to Real Adaptation of Pedestrian Detectors. *Sensors* **2020**, *20*, 5250. [CrossRef]