

Interactive Classification and Regression for Visual Tracking with Dual Update Strategy

Authors	Yuan, Di;Geng, Gu;Liu, Qiao;Chang, Xiaojun;He, Zhenyu
Citation	D. Yuan, G. Geng, Q. Liu, X. Chang, Z. He, "Interactive Classification and Regression for Visual Tracking with Dual Update Strategy," ACM Transactions on Multimedia Computing, Communications, and Applications, 2026, https://doi.org/10.1145/3803014 .
DOI	10.1145/3803014
Publisher	Association for Computing Machinery
Rights	Re-use licence for this version: In copyright
Download date	2026-06-07 05:02:54
Item License	http://rightsstatements.org/page/InC/1.0/
Link to Item	https://hdl.handle.net/20.500.14634/2352

Interactive Classification and Regression for Visual Tracking with Dual Update Strategy

DI YUAN and GU GENG, Guangzhou Institute of Technology, Xidian University, Guangzhou, China
QIAO LIU*, National Center for Applied Mathematics, Chongqing Normal University, China
XIAOJUN CHANG, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
ZHENYU HE, School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

The current two-stage tracking method locates the target using the position with the highest confidence score, and updates the template using a carefully designed template update strategy. However, we identify two key issues with these trackers: 1) the update strategy lacks continuous, cost-free template adaptation, leading to suboptimal tracking under appearance changes, and 2) the location with the highest confidence score does not always yield accurate bounding boxes, potentially resulting in incomplete target coverage. In this paper, we propose a novel tracker that incorporates two key innovations. First, the tracker employs a dual update strategy that performs online template updates at both the image and feature levels. This strategy enables continuous adaptation to target appearance changes without introducing additional computational overhead. Second, we enhance the existing loss function by introducing a Classification-Regression Interaction (CRI) loss, which guides the training process to produce confidence scores that more accurately reflect the quality of the predicted bounding boxes. Extensive experiments are conducted to evaluate the performance of our tracker and the effectiveness of the proposed methods. The experimental results show that our method has achieved a comprehensive improvement over the baseline on five datasets, and achieves competitive performance compared to state-of-the-art trackers.

CCS Concepts: • **Computing methodologies** → **Tracking**.

Additional Key Words and Phrases: Visual Tracking, Dual Update Strategy, Interaction Loss

1 Introduction

Visual object tracking task [1, 46, 51, 56] is to estimate the state of a specified target in subsequent frames based on the given target in the first frame. Visual object tracking has extensive applications in fields like autonomous driving, unmanned aerial vehicles, and human-computer interaction. However, adapting to changing targets presents significant challenges for the practical application of visual object tracking.

In recent years, template online updating strategies have been extensively explored in visual object tracking [7, 24, 25, 50]. Some trackers [17, 40, 48] adopt a straightforward replacement strategy, where the template is directly updated with the tracking result that has the highest confidence score. However, such approaches

*Corresponding author.

Authors' addresses: Di Yuan, dyuanhit@gmail.com, 23171214430@stu.xidian.edu.cn, Guangzhou Institute of Technology, Xidian University, Guangzhou, 510555, China; Qiao Liu, liuqiao.hit@gmail.com, National Center for Applied Mathematics, Chongqing Normal University, Chongqing, 401331, China; Xiaojun Chang, cxj273@gmail.com, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, 2007, Australia; Zhenyu He, zhenyuhe@hit.edu.cn, School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, 518055, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, or post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 1551-6865/2026/3-ART

<https://doi.org/10.1145/3803014>

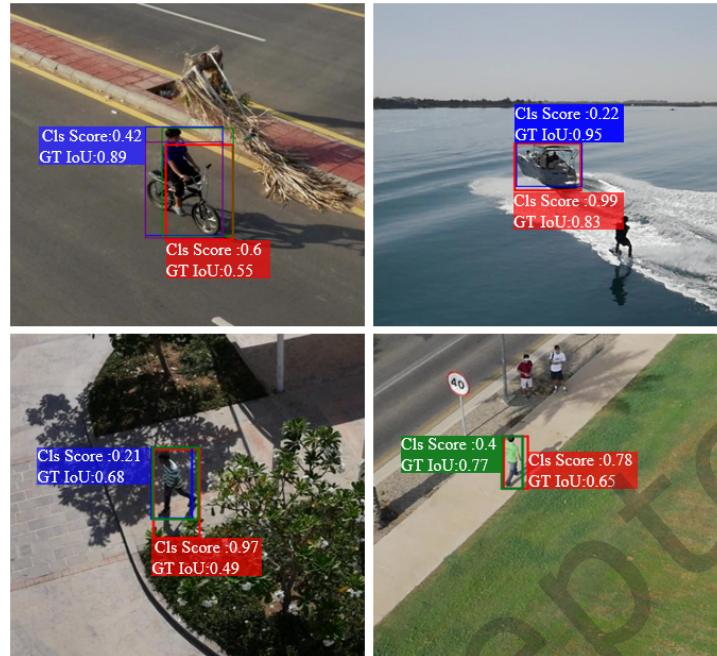


Fig. 1. Visualization of the predicted target confidence and its bounding box at the same location. In the figure, the Cls Score represents the target confidence predicted by the tracker, and GT IoU denotes the IoU between the predicted bounding box and the ground truth.

are highly sensitive to tracking errors—once a drift occurs, incorrect updates can quickly deteriorate tracking performance and robustness. To alleviate this issue, several studies [28] have proposed dual-template mechanisms that maintain two templates during tracking; one updated periodically at fixed intervals, and the other updated adaptively based on tracking confidence. Although these approaches are simple and effective under moderate conditions, they tend to fail in complex scenarios where the predefined update conditions are not satisfied, leading to stagnation of template updates and degraded long-term performance. To enhance adaptability, some methods [2, 21, 35, 54] perform online optimization of model parameters to cope with appearance variations of the target. While effective, such approaches incur substantial computational costs, significantly reducing tracking speed. Other methods [15, 19, 55] attempt to compress and dynamically update template features, which enables continuous adaptation to target changes. Nevertheless, feature compression may lead to the loss of discriminative information, thereby impairing tracking accuracy.

To address the above limitations, we propose a novel dual update strategy for visual tracking, which achieves continuous template adaptation without sacrificing real-time performance. Specifically, our tracker maintains two templates and updates them through two complementary mechanisms: Feature-level Update and Image-level Update. The Feature-level Update performs fine-grained and continuous adjustments in the feature space, ensuring persistent adaptation to subtle appearance changes. In contrast, the Image-level Update replaces the template at the image level to accommodate significant target variations and to correct potential drift introduced by the feature-level updates. By integrating these two complementary update schemes, the proposed method enables accurate and uninterrupted template adaptation during tracking, thereby enhancing the robustness and stability of the tracker in challenging scenarios.

On the other hand, the mainstream approaches in recent visual object tracking research are predominantly one-stage methods [14, 43, 55], which integrate feature extraction and feature interaction of both the template and the search region into a unified framework. This high degree of integration has led to significant improvements in tracking accuracy. However, such tightly coupled architectures make it challenging to incorporate flexible template update mechanisms to adapt to dynamically changing targets. In contrast, two-stage tracking methods [17, 25, 28] offer greater flexibility, allowing the integration of a wide range of effective template updating strategies. Typically, these methods output a target confidence score and a predicted bounding box for each spatial location within the search region, and the bounding box corresponding to the highest confidence score is selected as the final tracking result. Nevertheless, this approach relies on a critical assumption, that the predicted target confidence score accurately reflects the localization quality of the corresponding bounding box. In practice, we often observe inconsistencies between these two outputs, where a high confidence score may correspond to a low-quality bounding box (as illustrated in Fig. 1). Such discrepancies can lead the tracker to select inaccurate bounding boxes, thereby degrading overall tracking performance.

To address this issue, we optimize the training objective of the tracker by introducing an additional constraint term into the existing loss function. This design encourages mutual supervision between the classification and regression branches, enabling them to learn consistent representations during training. As a result, the predicted confidence scores become more representative of the actual localization accuracy, allowing the tracker to select bounding boxes more reliably. This optimization ultimately enhances both the tracking accuracy and robustness of the proposed framework.

In summary, our primary contributions are as follows:

- We propose a novel visual tracker that incorporates a dual template update mechanism capable of continuously adapting to target appearance changes without sacrificing tracking speed, thereby significantly improving tracking accuracy and robustness.
- We optimize the existing training objective by introducing a Classification–Regression Interaction Loss, which establishes mutual constraints between the classification and regression branches during training, enabling the tracker to generate more consistent and accurate bounding boxes.
- We conduct extensive experiments and ablation studies on five public tracking benchmarks, and the results clearly demonstrate the effectiveness of the proposed approach and its superior overall performance.

2 Related Work

Due to challenges such as target appearance variations, occlusion, illumination changes, and background clutter, visual tracking remains one of the most active and challenging topics in computer vision. In recent years, with the advent of deep learning and Transformer-based architectures, significant improvements have been achieved in both accuracy and robustness [16, 34, 49, 52]. However, differences in model design paradigms and update mechanisms still substantially affect the performance and adaptability of trackers. We only introduce the research progress that is most relevant to our work in three aspects: (1) Online update-based tracking methods, which dynamically adjust the template or model parameters during tracking to adapt to target appearance changes; (2) Two-stage visual object tracking methods, which separate the feature extraction and feature interaction processes to improve tracking precision through enhanced feature fusion; and (3) One-stage visual object tracking methods, which integrate feature extraction and interaction within a unified framework to achieve efficient and end-to-end learning. The following subsections provide a detailed overview and analysis of these three categories.

2.1 Online Update for Tracking

The state of objects in video sequences evolves over time, presenting significant challenges for long-term visual object tracking. In recent years, several trackers have addressed these challenges by employing carefully designed

online update strategies [2, 12, 45]. DiMP [2] continuously updates its model prediction network through an online iterative optimization process to accommodate target state changes. Liu et al. [24] proposed using the initial frame as the template under normal circumstances, but switching to the current frame as the template in the presence of background clutter. ToMP [25], STARK [42], and MixFormer [7] maintain two templates during tracking: one derived from the initial frame, which remains unchanged, and the other dynamically updated using traditional strategies. ROAM [4] utilizes an LSTM network for online fine-tuning, allowing for continuous model parameter updates. Similarly, LTMU [8] employs a three-stage cascading LSTM network to integrate semantic, appearance, and foreground-background classification information for guiding online updates of the tracker parameters. TrTr [53] introduces an additional online update module to correct the tracker’s foreground-background classification results. Although existing methods are designed cleverly, they often come with certain challenges. Some of these methods sacrifice tracking speed, while others fail to enable continuous online updates, leading to significant discrepancies between the template and the search region. In this paper, we propose a novel dual-update strategy. By leveraging this approach, we can achieve continuous template updates without any computational overhead, thus better adapting to changes in the target and ultimately enhancing the tracker’s performance.

2.2 Two-Stage Visual Object Tracking

The characteristic of two-stage visual object tracking methods is that the feature extraction and feature interaction processes are conducted separately. In the existing two-stage methods [11, 42, 44], the template and search region features are often extracted using CNN. The primary differences among various trackers lie in how these features are interacted and fused. Some approaches model the target’s appearance or background in the template and then use this model to classify foreground and background in the search region [3, 29]. Others, like TrTr [53] and SparseTT [11], enhance template features using self-attention mechanisms within transformers and then apply cross-attention mechanisms to query search region features over the template features. ToMP [25], similar to TrTr, employs a learnable query token to search for the target in the search region. TransT [6] designs a novel feature fusion module using a multi-head attention mechanism to merge template features with search region features. Although the feature interaction methods in existing approaches are distinctive, they generally adopt a similar strategy for predicting the final bounding box: for each position, a foreground-background classification confidence and a bounding box are predicted, and the bounding box corresponding to the position with the highest classification confidence is selected as the final prediction. However, we observe that the position with the highest classification confidence does not necessarily correspond to the most accurate bounding box regression, which can result in the tracker losing the most accurate bounding box. To address this issue, we propose guiding the foreground-background classification process and the bounding box regression process mutually, so that the target confidence score predicted by the tracker more accurately reflects the actual accuracy of the bounding box, thereby improving the accuracy of bounding box selection.

2.3 One-Stage Visual Object Tracking

The one-stage visual object tracking methods [14, 43, 55] have demonstrated superior tracking performance by integrating the processes of feature extraction and feature interaction between the template image and the search region. Among existing one-stage methods, OTrack [43] is the first to combine feature extraction and feature interaction, achieving more accurate and faster tracking by eliminating irrelevant tokens from the search region that are not related to the template. Subsequently, GRM [14] and AVTrack [20] improve the tracking performance over OTrack by optimizing the attention mechanism between the template and the search region. ARTrack [39] and ARTrackV2 [1] significantly enhance tracking performance by generating object bounding boxes in an autoregressive manner. LoRAT [22], by introducing the Parameter-Efficient Fine-Tuning technique from large model training, reduces the hardware requirements for training the tracker and makes it feasible to

perform tracking with larger pre-trained models. Despite the remarkable success of these methods, the highly integrated nature of one-stage methods prevents them from utilizing online template update methods to adapt to object changes. To address this limitation, ODTrack [55] introduces a token that spans the entire tracking process, continuously compressing target information into this token during tracking to help the tracker adapt to changes in the target. However, this compression may lead to the loss of key features of the target. In this paper, we propose a two-stage method that incorporates our novel dual update template online update strategy, allowing the tracker to better adapt to object changes.

3 Method

In this paper, we first review the existing method [25], which serves as our baseline. Building upon this foundation, we integrate it with our proposed approach to develop a more advanced tracker, named CRToMP. The overall architecture of CRToMP is illustrated in Fig. 2. Specifically, the input templates t_1 and t_2 , together with the search region t_s , are first processed to extract image features. These features are then fed into a Transformer encoder to perform feature interaction, followed by a Transformer decoder that conducts target querying. Based on the decoded query representations, the tracker predicts the final target bounding boxes. The output \hat{y}_s represents the foreground-background classification confidence scores for each spatial location, while \hat{d}_s denotes the predicted target bounding box in the left-top-right-bottom (ltrb) format. Other variables correspond to intermediate feature representations generated during the feature extraction and interaction stages, which will be discussed in detail in the subsequent sections. First, we review the fundamental process of previous method in Section 3.1. Next, we provide a detailed description of the dual update strategy in Section 3.2. Finally, we explain how to optimize the loss function to simultaneously constrain and enable mutual learning between the foreground-background classification process and the bounding box regression process during the training of the tracker in Section 3.3.

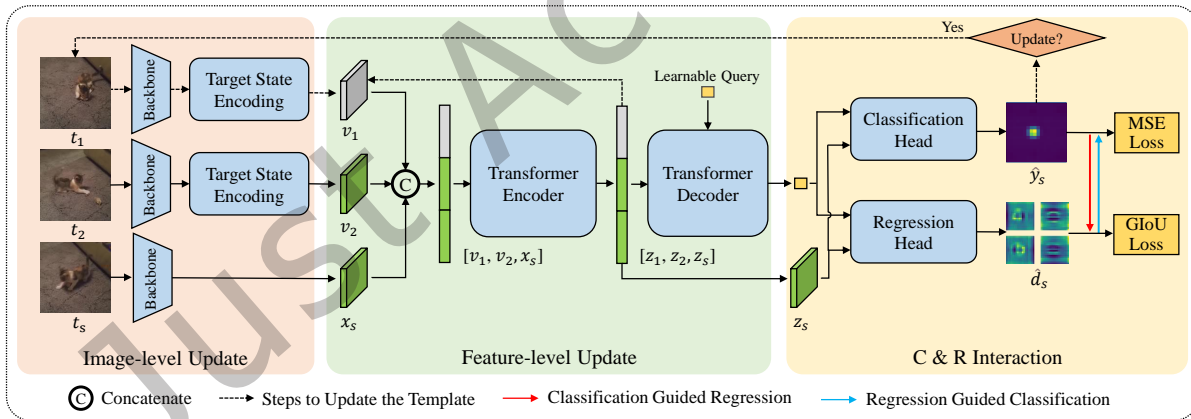


Fig. 2. Overview of our proposed CRToMP. In the tracking process, CRToMP performs image-level update on the template frame and feature-level update on the features of the template. The two processes together constitute a dual update strategy. In the training process, the classification network and the regression network weight each other's loss calculation to achieve mutual alignment of the two networks. Solid lines indicate the tracking steps and dashed lines represent the online update steps.

3.1 Method Review

In this section, we review the previous method [25] that primarily consists of six components: backbone F , target state encoding, transformer encoder T_{enc} , transformer decoder T_{dec} , classification network H_{cls} , and bounding box regression network H_{bbox} . During tracking, tracker takes two template images, t_1 and t_2 , and a search region image t_s , and then outputs the predicted target confidence score \hat{y}_s and the target bounding box \hat{d}_s .

Specifically, tracker first extracts the features of two template images and the search region image through backbone F :

$$x_i = F(t_i) \quad i = 1, 2, s. \quad (1)$$

Subsequently, in the target state encoding, tracker integrates the bounding box information d and positional information y of the target from the template into the template features x :

$$v_i = x_i + \phi(d_i) + y_i \cdot e_{fg}, \quad i = 1, 2, \quad (2)$$

where $\phi(\cdot)$ represents a four-layer *MLP*, and e_{fg} is a learnable vector, with \cdot denoting element-wise multiplication. Next, tracker concatenates the search region image feature map x_s with the fused template features v_1 and v_2 along the spatial dimension. These concatenated features are then input into the transformer encoder T_{enc} for feature interaction:

$$[z_1, z_2, z_s] = T_{enc}([v_1, v_2, x_s]), \quad (3)$$

where T_{enc} represents six stacked normal self-attention transformer blocks. Next, tracker utilizes a learnable vector e_{fg} as a query vector to query the target within the features after the interaction:

$$\omega = T_{dec}([z_1, z_2, z_s], e_{fg}), \quad (4)$$

where T_{dec} represents six stacked normal cross-attention transformer blocks. Finally, the queried vector ω is combined with the interacted search region feature z_s and input into the classification network and the regression network for foreground-background classification and bounding box regression, respectively:

$$\hat{y}_s = H_{cls}(\omega, z_s), \quad (5)$$

$$\hat{d}_s = H_{bbox}(\omega, z_s), \quad (6)$$

where $\hat{y}_s \in h \times w$ and $\hat{d}_s \in h \times w \times 4$, with h and w representing the height and width of the search region image feature map, respectively.

Additionally, during training, the classification loss for previous tracker can be expressed as follows:

$$L_{cls} = \sum_{m=0}^h \sum_{n=0}^w \|l_{cls}(\hat{y}_s^{m,n}, y_s^{m,n})\|^2, \quad (7)$$

where $\hat{y}_s^{m,n}$ represents the predicted target confidence score at position (m, n) , and h and w are the height and width of the search region image feature map. $y_s^{m,n}$ is the ground truth, representing the true target confidence score, which is generated by a Gaussian function. $l_{cls}(\cdot)$ is a function from DiMP [2] that includes different losses for the background and foreground regions. The regression loss used by tracker can be expressed as:

$$L_{giou} = \sum_{m=0}^h \sum_{n=0}^w (1 - l_{giou}(\hat{d}_s^{m,n}, d_s^{m,n})), \quad (8)$$

where $\hat{d}_s^{m,n}$ represents the predicted *ltrb* bounding box representation by the tracker at position (m, n) , and $d_s^{m,n}$ denotes the corresponding ground truth values. $l_{giou}(\cdot)$ is the Generalized Intersection over Union loss [30], which is calculated based on the *ltrb* bounding box representation. For more detailed information about the *ltrb*

representation, please refer to Section 3.3 of the ToMP [25] reference. In summary, the total loss of previous tracker during training can be expressed as:

$$L_{pre} = \lambda L_{cls}(\hat{y}_s, y_s) + L_{giou}(\hat{d}_s, d_s), \quad (9)$$

where λ is a scalar weighting the contribution of loss and set to 100.

3.2 Dual Update Strategy

The state of target in a video sequence continuously evolves over time. To effectively adapt to these dynamic changes, we propose an online template update strategy called dual update strategy. As illustrated in Fig. 2, this approach consists of two components: image-level update and feature-level update. The image-level update involves replacing the template t_1 at the image level according to a specific strategy, while the feature-level update continuously updates the feature of the template t_1 . In the following, we will provide a detailed explanation of these two components and then integrate them into the final dual update strategy.

Image-level Update. As shown in the image-level of Fig. 2, we maintain two templates during the tracking process. Template t_2 is initialized with the first frame and remains fixed to ensure accurate tracking. The other template, t_1 , is updated based on the confidence of the tracked results. The confidence of the tracked results is assessed by the target confidence score predicted by the tracker. If the confidence of the current tracked result is denoted as C , then we have:

$$C = \max(\hat{y}_s), \quad (10)$$

where $\hat{y}_s \in h \times w$ is target confidence score predicted by the tracker. Template t_1 is updated to the current frame only when the confidence C of the current tracked result exceeds a given threshold T_1 . Therefore, the image-level update process can be expressed as:

$$\begin{cases} t_1 = S_i, t_2 = S_1 & \text{if } C_i > T_1 \\ t_1 = t_1, t_2 = S_1 & \text{otherwise} \end{cases} \quad (11)$$

where S_i denote the i -th video frame, and C_i represent the confidence of the tracked result for the i -th video frame. Through the image-level update, we can accurately update both the appearance of the target and the background in the template. However, this strategy fails to continuously update the template. Once the tracking confidence decreases, the template update process stagnates. Although lowering the threshold T_1 can address this issue, it increases the risk of updating the template with an incorrect target, which in turn reduces the tracker's accuracy.

Feature-level Update. To continuously update the template in response to changes in the target, we propose the feature-level update. Drawing inspiration from the time token concept in ODTrack [55], we leverage the self-attention mechanism in the transformer to capture temporal changes in the target. However, unlike ODTrack, which introduces an additional token to compress and store this change information, we directly update the features of the template to avoid the loss of key target features during the compression process. The specific process is shown in the feature-level update section of Fig. 2. In brief, we update the feature of template t_1 to the feature output by the transformer encoder. By performing self-attention calculations with the search region image, we update the feature of template t_1 , ensuring its similarity to the current target. The entire process can be expressed as follows:

$$[z_1^j, z_2^j, z_s^j] = T_{enc}([z_1^{j-1}, v_2^j, x_s^j]), \quad (12)$$

where the superscript j denotes the tracking process of the j -th video frame in video sequence, where $j \geq 1$. When $j = 1$, $z_1^0 = v_2^1$. Additionally, in order to force the transformer encoder to be able to capture the temporal variation of the target, we use the process described in equation (12) during training to perform continuous tracking across multiple frames. As a result, through feature-level updates, we enable the tracker to continuously adapt to

Algorithm 1: Dual Update Strategy

Input: Templates t_1, t_2 , Video Sequence $S = \{S_i\}_{i=2}^n$

```

1 Memory  $\leftarrow [t_1]$ ;
2 for  $t_s \leftarrow S_2$  to  $S_n$  do
3    $\hat{y}_s, \hat{d}_s \leftarrow \text{Track}(t_1, t_2, t_s)$ ; // feature-level
4   if  $\max(\hat{y}_s) > T_1$  then
5     | Memory.append( $t_s$ );
6   end
7   if  $\max(\hat{y}_s) < T_2$  then
8     |  $t_1 \leftarrow \text{Memory}[-1]$ ; // image-level
9   end
10 end

```

changes in the target without any additional cost. However, due to the uncertainty in tracking confidence, during low-confidence tracking, feature-level updates are highly likely to update the feature of template t_1 to the wrong target.

Dual Update Strategy. In this section, we design a strategy to integrate image-level and feature-level updates, enabling the two components to complement each other. The specific process of the dual update strategy is outlined in Algorithm 1. Initially, the input templates t_1 and t_2 are set as the first frame of the video. During the algorithm execution, we control the update of template t_1 by setting two thresholds, T_1 and T_2 , where T_1 governs template storage and T_2 governs template correction. On one hand, when the tracking confidence exceeds T_1 , it indicates that the template's accuracy is high, and the template is stored in memory. On the other hand, when the tracking confidence drops below T_2 , it indicates that the template's feature has been updated in the wrong direction, so the accurate template is retrieved from memory to reinitialize the template feature. It is noteworthy that we do not directly update template t_1 when the tracking confidence exceeds T_1 because we believe the feature-level update is more effective at maintaining the similarity between the template and the current target. Through the dual update strategy, we achieve continuous and accurate online updates of the template without additional cost, along with automatic correction functionality.

Differences from Related Template Update Approaches. The proposed dual update strategy is characterized by its ability to continuously update template features while adaptively correcting erroneous updates. Although some existing methods [7, 25, 28] also utilize two templates, they typically rely on fixed-interval updates or conditional triggers rather than continuous adaptation. As a result, these methods may fail to update the template in a timely manner when the target undergoes significant appearance changes. In contrast, the feature-level update in our approach enables uninterrupted adaptation to such variations. Other strategies [19, 47, 55] achieve continuous adaptation by dynamically updating template features, however, when the target temporarily disappears or occludes, these methods often lead to feature drift by updating in the wrong direction. Unlike these approaches, our method incorporates an image-level update mechanism that effectively corrects such erroneous updates, thereby ensuring the reliability and correctness of the template updates.

3.3 Classification-Regression Interaction Loss

Previous two-stage trackers perform bounding box regression at the location with the highest target confidence score. However, we find that the location with the highest target confidence score is not necessarily the location of the most accurate bounding box. To address this issue, we propose a training process where the classification

Table 1. Comparison of model parameters, MACs, and inference speed on LaSOT.

Method	Backbone	Resolution	Params	MACs	Speed	Device	AUC
Baseline	ResNet50	288×288	25.7M	26.1G	76 fps	3090	67.6%
CRToMP	ResNet50	288×288	25.7M	26.1G	77 fps	3090	69.4%
CRToMP-iTPN	Fast-iTPN	288×288	81.7M	80.0G	30 fps	3090	71.6%

results and regression results mutually guide each other to achieve alignment. Specifically, we optimize the existing classification loss and regression loss, this process is illustrated in the C&R Interaction part of Fig. 2.

For the regression loss, we base our approach on a simple idea: *if the predicted target confidence score is high, the corresponding predicted bounding box should also be accurate*. Therefore, at each position, we weight the regression loss using the predicted target confidence score:

$$L_{giou}^{CR} = \sum_{m=0}^h \sum_{n=0}^w l_{giou}(\hat{d}_s^{m,n}, d_s^{m,n}) \cdot w_{reg}^{m,n}, \quad (13)$$

$$w_{reg}^{m,n} = (1 + \max(\hat{y}_s^{m,n}, 0)). \quad (14)$$

The detailed meanings of the variables in Eq. (13) can be found in Section 3.1, and will not be repeated here. By assigning a larger contribution weight to the regression loss at positions with higher target confidence score, we force positions with high confidence score to correspond to more accurate predicted bounding boxes. Similarly, for the classification loss, we use giou between the predicted bounding box and the ground truth as the contribution weight for the classification loss, guiding the learning of the classification network:

$$L_{cls}^{CR} = \sum_{m=0}^h \sum_{n=0}^w \|l_{cls}(\hat{y}_s^{m,n}, y_s^{m,n})\|^2 \cdot w_{cls}^{m,n}, \quad (15)$$

$$w_{cls}^{m,n} = (1 + \max(l_{giou}(\hat{d}_s^{m,n}, d_s^{m,n}), 0)). \quad (16)$$

By assigning a larger contribution weight to the classification loss at positions where the predicted bounding box is more accurate, we force accurate bounding boxes to have corresponding accurate target confidence scores. Finally, the total loss is expressed as:

$$L_{CR} = \lambda L_{cls}^{CR} + L_{giou}^{CR}, \quad (17)$$

where λ is a scalar weighting the contribution of the loss. In this paper, we set it as 100, consistent with baseline.

4 Experiment

4.1 Implementation Details

We present two variants of our proposed model, as summarized in Table 1, and conduct a detailed comparison with the baseline. CRToMP is designed to share the same backbone as the baseline in order to more clearly demonstrate the effectiveness of our update strategy. CRToMP-iTPN is obtained by replacing the backbone of CRToMP with Fast-iTPN [36] for highlighting the competitiveness of our tracker.

During offline training, the input sizes of both the template and search regions were set to 288. We trained our tracker on the CoCo [23], TrackingNet [27], GOT-10k [18], and LaSOT [10] datasets. We use typical commonly used data augmentation methods, including horizontal flipping and brightness jittering. We train CRToMP with AdamW optimizer, set the weight decay to 10^{-4} , the initial learning rate for the prediction head network to 2×10^{-5} , and other parameters to 4×10^{-6} . We trained the model for a total of 300 epochs, using a total of 40k

Table 2. Validation of the proposed method, experiments are conducted on LaSOT, TrackingNet and LaSOT_{ext}.

DUS	CRI	LaSOT(%)	TrackingNet(%)	LaSOT _{ext} (%)	Speed(fps)	MACs(G)
		67.6	80.7	45.4	76	26.1
✓		68.3	81.9	47.2	77	26.1
✓	✓	69.4	81.9	49.1	77	26.1

image pairs. At the 150th epoch and 250th, the learning rate was reduced to one-fifth. The model is conducted on a server with a 24GB RTX 3090GPUs, using a batch size of 32, where each batch consists of three search image and two template images. During the testing process, parameter T_1 was set to 0.85 and parameter T_2 was set to 0.75.

4.2 Ablation Studies

The effectiveness of both methods. To demonstrate the effectiveness of the dual update strategy and the classification-regression interaction loss, we conducted ablation studies on these methods using the LaSOT [10], TrackingNet [27], and LaSOT_{ext} [9] benchmarks. The specific experimental results are shown in Table 2. In the table, "DUS" denotes the dual update strategy, and "CRI" denotes the classification-regression interaction loss. Compared to the baseline, the use of the dual update strategy increased the AUC by 0.7% on LaSOT, by 1.2% on TrackingNet and by 1.8% on LaSOT_{ext}. This shows that the dual update strategy provides more accurate target state information than the original strategy. Based on the dual update strategy, the use of classification-regression interaction loss further improved the AUC on LaSOT by another 1.1% and by 1.9% on LaSOT_{ext}. This shows that classification-regression interaction loss can indeed help the tracker select more accurate bounding boxes.

In addition to accuracy, the experimental results in the Table 2 also show that the tracker's speed did not decrease after applying our method, and there was even a slight improvement of 1 FPS. This is attributed to the fact that our proposed approach is integrated into the template update and offline training stages, thereby incurring no additional computational overhead during online tracking. For changes in speed, We hypothesize that this improvement may be due to the increased update threshold T_1 during template updates, which reduced the frequency of storing the latest template and thereby slightly improved the speed. In summary, the dual update strategy and the classification-regression interaction loss proposed in this paper both effectively enhance the tracker's accuracy, and when combined, they further improve performance.

The effectiveness of the two update processes. To verify that both update processes of the dual update strategy are effective, we conducted ablation studies on the baseline equipped with the image-level update and feature-level update processes on the LaSOT [10] benchmark. The specific experimental results are shown in Table 3. From the experimental data in the Table 3, we can see that when the image-level update process is added alone, the AUC improves by 0.6% and the Precision improves by 0.8%. When the feature-level update process is added alone, there is a minor improvement in tracking performance, with AUC and Precision increasing by 0.3% and 0.5%, respectively. Finally, when using the dual update strategy, AUC and Precision are further improved upon the image-level update. These data indicate that both update processes in the dual update strategy are effective.

The sensitivity analysis of thresholds in dual update strategy. To analyze the sensitivity of the thresholds T_1 and T_2 in the dual update strategy, we conducted a series of experiments on the LaSOT [10] dataset using different threshold combinations, as summarized in Table 4. It is noted that T_1 should always be greater than T_2 to ensure that the image-level update can effectively correct potential deviations introduced during the feature-level update. As shown in Table 4, the tracker achieves the best performance when $T_1 = 0.85$ and $T_2 = 0.75$, with an

Table 3. The ablation experiments on LaSOT to verify effectiveness of the two update processes.

Image-level	Feature-level	AUC(%)	Precision(%)
		67.5	72.4
✓		68.1	73.2
	✓	67.8	72.9
✓	✓	68.3	73.4

Table 4. Performance comparison of trackers with different thresholds in the dual update strategy on the LaSOT.

T_1	T_2	AUC(%)	Precision(%)
0.90	0.80	69.0	74.3
0.85	0.80	69.2	74.2
0.90	0.75	69.3	74.6
0.85	0.75	69.4	74.6
0.80	0.75	69.0	74.2
0.90	0.70	69.0	74.1
0.85	0.70	69.0	74.1
0.80	0.70	69.1	74.3

AUC of 69.4% and a Precision of 74.6%. This indicates that the chosen thresholds provide a favorable balance between adaptability and stability. When T_1 increases further (e.g., to 0.90), the image-level update becomes too conservative due to a higher activation threshold, resulting in reduced update frequency and degraded adaptability. Conversely, a smaller T_1 triggers overly frequent updates, increasing the risk of incorporating inaccurate templates and thus decreasing tracking accuracy. Regarding T_2 , setting it too high (e.g., 0.80) causes frequent resets of the feature-level update, which undermines its effectiveness. In contrast, a too-small T_2 prevents timely image-level corrections when tracking drift occurs, reducing the overall robustness of the model. In summary, the combination of $T_1 = 0.85$ and $T_2 = 0.75$ yields the best trade-off between precision and robustness, demonstrating its effectiveness as the optimal parameter configuration for the dual update strategy.

Analysis in complex scenarios. To evaluate the performance of our method under various challenging scenarios, we categorized the videos in the LaSOT dataset according to different challenge attributes, including Camera Motion (CM), Motion Blur (MB), Deformation (D), Partial Occlusion (PO), Illumination Variation (IV), Aspect Ratio Change (ARC), Low Resolution (LR), Out-of-View (OV), Fast Motion (FM), Full Occlusion (FO), Scale Variation (SV), Viewpoint Change (VC), Background Clutter (BC), and Rotation (R). We then evaluated both the baseline tracker and our proposed methods on each attribute category to analyze the effectiveness and robustness of the proposed approach across diverse tracking conditions. The evaluation results are presented in Table 5 and Fig. 3. Compared to the baseline, the tracker with the proposed dual template update strategy shows significant improvements in challenging scenarios such as deformation, camera motion, motion blur, and partial occlusion. This indicates that the dual template update mechanism enables better adaptation to target appearance variations. Moreover, after incorporating the Classification-Regression Interaction loss, CRTOMP exhibits consistent performance improvements over the baseline across all attribute categories. In particular, tracking accuracy in challenging conditions such as Motion Blur, Low Resolution, and Fast Motion is further enhanced on top of the dual update strategy. Notably, in Viewpoint Change and Out-of-View scenarios, the

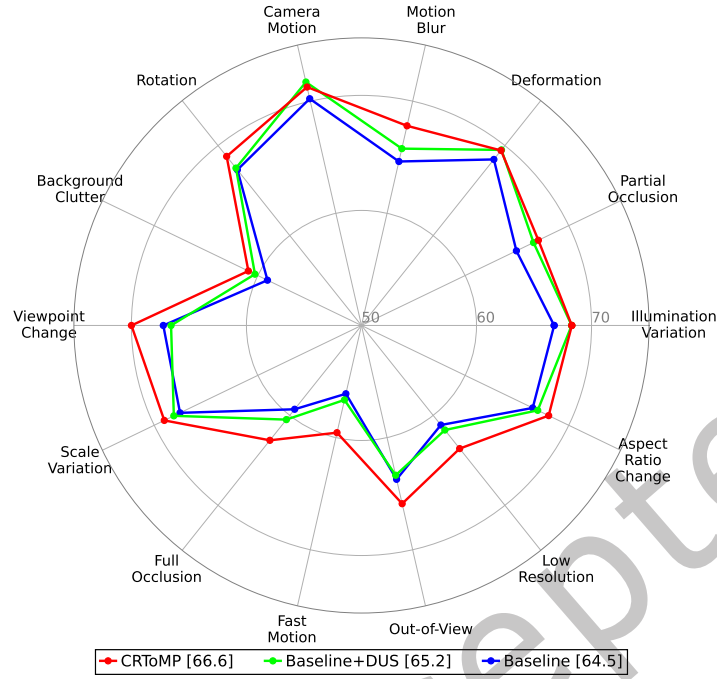


Fig. 3. Effectiveness of the proposed method in various challenge scenarios in LaSOT.

Table 5. Effectiveness of the proposed method in various challenge scenarios in LaSOT, and challenge scenarios use the first letter abbreviation.

Method	IV	PO	D	MB	CM	R	BC	VC	SV	FO	FM	OV	LR	ARC
Baseline	66.8	65.0	68.5	64.6	70.2	67.3	59.1	67.2	67.5	59.3	56.1	63.7	61.1	66.5
Baseline+DUS	68.3	66.6	69.5	65.8	71.7	67.5	60.3	66.6	68.1	60.5	56.6	63.4	61.3	67.0
CRToMP	68.3	67.1	69.5	67.8	71.3	68.8	60.9	70.0	69.0	62.8	59.6	65.9	63.8	68.3

tracker’s ability to re-detect the target significantly improves, benefitting from the increased prediction accuracy introduced by the CRI loss. This improvement suggests that the retrained tracker produces a more consistent relationship between the predicted classification confidence and the localization accuracy of bounding boxes. Consequently, the bounding box selected based on the highest confidence score more accurately represents the true target location. Overall, the proposed method demonstrates strong robustness and generalization ability across diverse and challenging tracking scenarios.

Comparison with other update strategies. To demonstrate the effectiveness of our proposed template update strategy, we compare it against commonly used update methods, as shown in the table 6. Method ① directly updates templates with high-confidence predictions, results in a 0.5% drop in AO. This is likely because challenging scenarios may lower tracking confidence, leading to stalled template updates. Method ② updates templates at fixed frame intervals, causes a 0.9% drop in AO, indicating that continuous template updates are beneficial for tracking accuracy. Method ③ continuously updates template features without correction, leads to a

Table 6. Comparison of Different Update Strategies on GOT-10K. ①: update with high-confidence results; ②: update at fixed intervals; ③: continuous feature updating.

Method	AO(%)	$SR_{0.5}(\%)$	$SR_{0.75}(\%)$
Ours	72.9	83.9	68.6
①	72.4	83.4	68.4
②	72.0	82.8	67.9
③	72.5	83.4	68.5

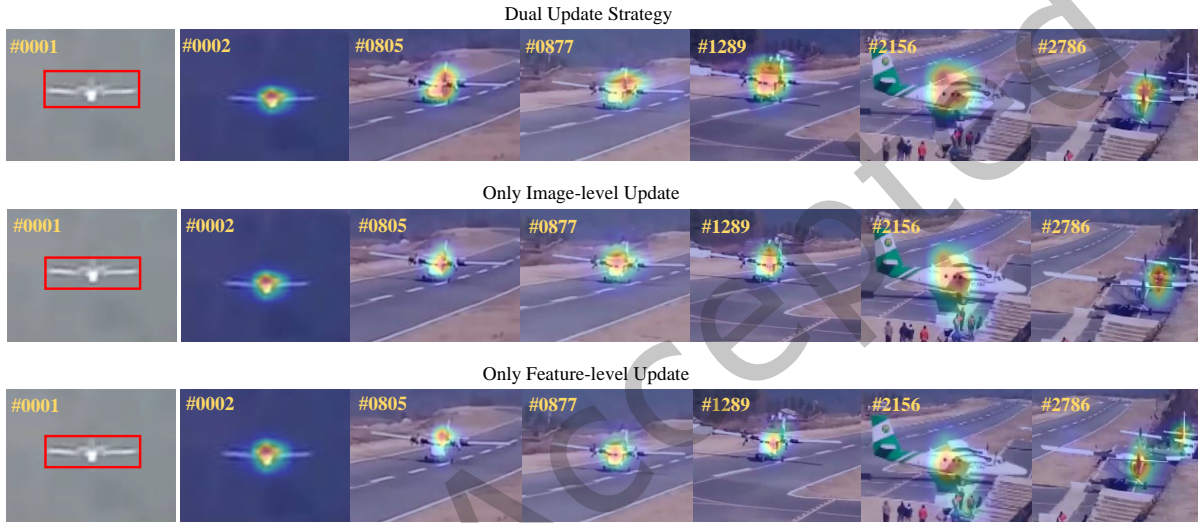


Fig. 4. Visualization of the attention map between template t_1 and the search region t_s . The first column is the specified target template, the other columns are the search region, and the redder color represents the greater attention value between the template and the search region.

0.4% decrease in AO, highlighting the importance of the error-correction mechanism in our image-level update strategy.

4.3 Visualization of Attention Map

In this section, to further understand the roles of the two updates in the dual-update strategy, we visualized the attention map of template t_1 on the search region t_s in the final layer of the Transformer Encoder. Fig. 4 shows visualizations for three different scenarios: using the dual-update strategy, using only the image-level update, and using only the feature-level update, allowing for a comparison of the effects of each component. From the figure, it can be observed that, compared to the first row, the attention in the second and third rows is initially focused on the target at the beginning of the sequence. However, as the target undergoes significant changes, the attention of the template in the search region gradually starts to disperse, as seen in frame 2156. In frame 2786, the attention in the second row shifts within the search region, while in the third row, the attention even disperses to the wrong object. This demonstrates that the dual update strategy effectively integrates feature-level

Table 7. Comparison on LaSOT, GOT-10k and TrackingNet. The best two results are highlighted in red and blue, respectively. The underline indicates the best result among the trackers of the current category. * denotes we reproduce the results using the officially published model weights.

Method	LaSOT			GOT-10k			TrackingNet			
	AUC(%)	P_{Norm} (%)	P(%)	AO(%)	$SR_{0.5}$ (%)	$SR_{0.75}$ (%)	AUC(%)	P_{Norm} (%)	P(%)	
One-Stage	OTrack ₂₅₆ [43]	69.1	78.7	75.2	71.0	80.4	68.2	83.1	87.8	82.0
	MixFormer-22k[7]	69.2	78.7	74.7	70.7	80.0	67.8	83.1	88.1	81.6
	EVPTrack ₂₂₄ [32]	70.4	<u>80.9</u>	77.2	73.3	83.6	70.7	83.5	88.3	–
	ROMTrack ₂₅₆ [4]	69.3	78.8	75.6	72.9	82.9	70.2	83.6	88.4	82.7
	SeqTrack ₂₅₆ [5]	69.9	79.7	76.3	<u>74.7</u>	<u>84.7</u>	71.8	83.3	88.3	82.2
	LATrack ₂₅₆ [31]	69.7	79.1	75.5	74.2	84.0	70.4	83.7	<u>88.7</u>	82.5
	MCTrack ₂₅₆ [38]	69.2	78.3	74.1	73.6	83.7	71.4	82.5	–	–
	ARTrack-B ₂₅₆ [39]	70.4	79.5	76.6	73.5	82.2	70.9	<u>84.2</u>	<u>88.7</u>	<u>83.5</u>
	DiffusionTrack-B ₂₅₆ [41]	<u>70.8</u>	79.8	<u>76.7</u>	<u>74.8</u>	<u>85.4</u>	<u>72.0</u>	83.8	88.2	82.1
Two-Stage	AiATrack[13]	69.0	79.4	73.8	69.6	80.0	63.2	82.7	87.8	80.4
	STARK[42]	67.1	77.0	–	68.8	78.1	64.1	82.0	86.9	–
	HAT[37]	66.3	74.3	69.4	–	–	–	80.7	85.0	78.1
	SparseTT[11]	66.0	74.8	70.1	69.3	79.1	63.8	81.7	86.6	79.5
	PromptVT[12]	63.7	73.8	66.8	68.2	79.3	61.8	78.0	83.5	74.4
	CSWinTT[33]	66.2	75.2	70.9	69.4	78.9	65.4	81.9	86.7	79.5
	TransT[6]	68.0	76.9	72.4	67.7	77.1	61.5	82.1	87.2	80.4
	ToMP50[25]	67.6	78.0	72.2	–	–	–	81.2	86.2	78.6
	ToMP50*	67.6	78.0	72.2	72.4	<u>84.2</u>	66.2	80.7	85.6	78.1
	CRToMP(Ours)	69.4	79.3	74.6	72.9	83.9	68.6	81.9	86.4	79.7
	CRToMP-iTPN(Ours)	<u>71.6</u>	<u>80.6</u>	<u>77.6</u>	<u>74.8</u>	83.7	<u>73.4</u>	<u>84.6</u>	<u>88.8</u>	<u>84.3</u>

and image-level updates, allowing both components to complement each other and achieve better performance. This finding is consistent with the conclusions drawn from the ablation study part.

4.4 Comparisons with other trackers

To demonstrate that our tracker is competitive compared to other trackers, we compared our tracker with others on five benchmarks. The specific results are shown in Table 7 and Table 8. To gain a clearer understanding of the competitiveness of our tracker, we categorized the compared trackers into two groups in Table 7: based purely on Transformers and based on CNN-Transformers.

LaSOT_{ext}[10]: LaSOT_{ext} is an extension of the LaSOT test set. It comprises 150 video sequences and 15 distinct target categories that have no overlaps with those in the LaSOT dataset. The evaluation results on LaSOT_{ext} are shown in Table 8. It can be seen that CRToMP outperforms ToMP50 by 3.2% in AUC and CRToMP-iTPN achieves competitive tracking performance among all trackers.

Table 8. The performance of our method and other methods on LaSOT_{ext} and UAV123 in terms of AUC. The best two results are highlighted in red and blue. * denotes we reproduce the results using the officially published model weights.

Method	LaSOT _{ext}	UAV123
OTrack ₂₅₆ [43]	47.4	68.3
SeqTrack ₂₅₆ [5]	49.5	69.2
ARTrack ₂₅₆ [39]	46.4	67.7
LATrack ₂₅₆ [31]	48.7	69.2
MCTrack ₂₅₆ [38]	47.4	68.5
ToMP50[25]	45.4	69.0
ToMP50*	45.4	68.0
CRToMP(Ours)	49.1	68.6
CRToMP-iTPN(Ours)	49.6	69.4

LaSOT[10]: LaSOT is a large-scale long-term video dataset. Its test split contains 280 long videos, with an average length exceeding 2500 frames. We evaluated the long-term tracking performance of the tracker on the test split of LaSOT, and the experimental results are shown in Table 7. From the table, it can be seen that our tracker CRToMP outperforms ToMP50 by 1.8% in AUC. Moreover, CRToMP-iTPN achieves the best performance among all trackers and CRToMP also shows strong competitiveness. This indicates that our tracker has excellent long video tracking capabilities.

TrackingNet[27]: The test split of TrackingNet comprises 511 video sequences, featuring a rich variety of target categories and video scene. We assessed the tracking performance of our tracker across various scenarios on the TrackingNet test set, with the results presented in Table 7. As can be seen from the table, CRToMP-iTPN achieves the best performance among our model variants and performs on par with the top-performing single-stage trackers, demonstrating that our tracker attains competitive performance in complex scenarios.

Got-10k[18]: The test split of Got-10k includes 180 video sequences which contain target categories not present in the training data. It is used to evaluate the generalization capability of trackers. The evaluation results on the test split of Got-10k are shown in Table 7. It can be observed that CRToMP-iTPN achieved the best performance among all trackers, indicating that our tracker possesses strong generalization capabilities.

UAV123[26]: UAV123 is a low-altitude aerial dataset captured by drones which comprises 123 sequences. We evaluate the robustness of the trackers against continuously changing viewpoints and small target on the UAV123 dataset and the evaluation results are shown in Table 8. From the table, it can be seen that CRToMP outperforms ToMP50 by 0.6%, and CRToMP-iTPN achieves the best performance. This indicates that our method still possesses good tracking performance for small targets.

4.5 Qualitative Study

In Fig. 5, we present a visual comparison of the tracking results of different trackers after long-term tracking. This includes scenarios such as occlusion, deformation, similar targets, scale variation, and background interference. The first column shows the specified tracking target in the template, and the subsequent columns display the visualization of tracking results on different video frames. It can be observed that, compared to other trackers, CRToMP exhibits better robustness when dealing with these complex situations. Especially, the comparison with the tracking results of ToMP demonstrates that our method can enhance the robustness of the tracker when facing these complex situations.

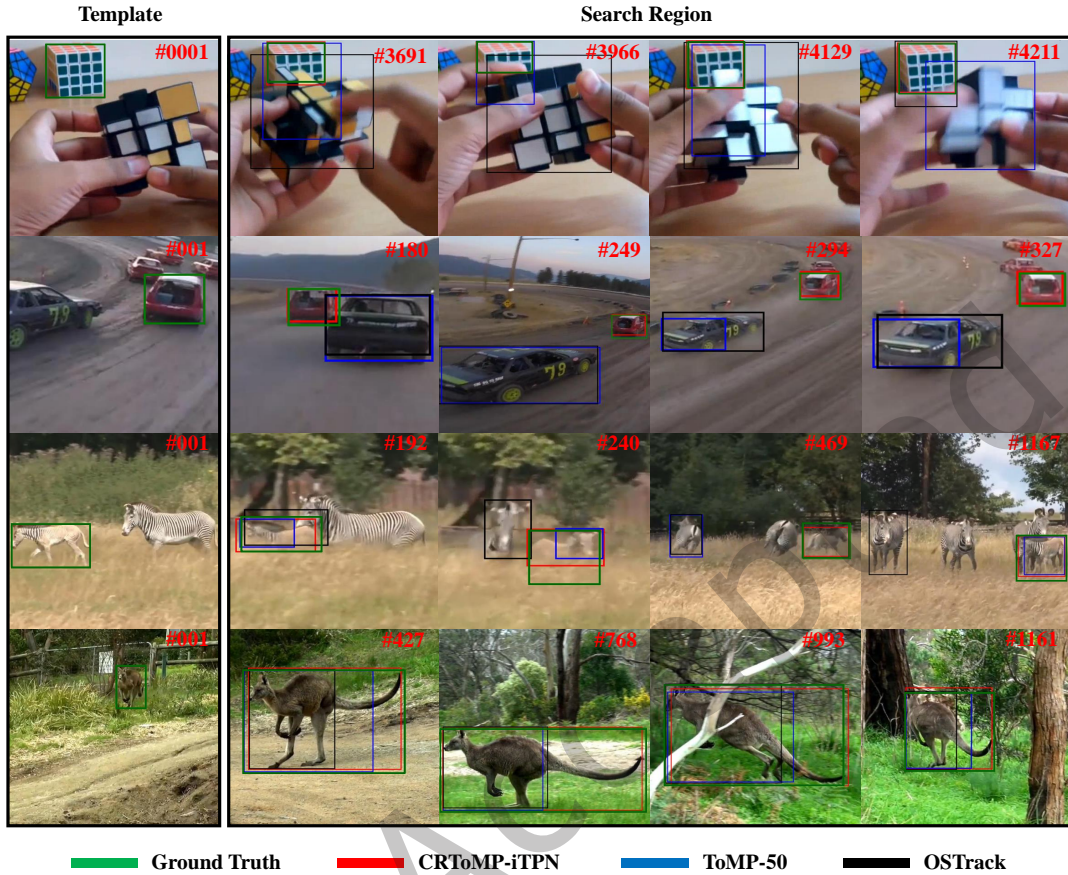


Fig. 5. Visualization results of tracking processes for multiple trackers. Zoom in for a clearer view.

4.6 Limitations Analysis and Future Work

Despite the significant improvements achieved in both tracking accuracy and efficiency, our method is not without limitations. As shown in Fig. 3, the tracker exhibits relatively lower robustness in scenarios involving fast target motion and highly cluttered backgrounds. This is further illustrated in Fig. 6, where we compare our tracker with state-of-the-art one-stage methods in challenging cases. In the first and second rows, which depict scenes with visually similar distractors in the background, the tracker is prone to drifting towards incorrect targets. In the third row, where the target undergoes rapid motion, the tracker again struggles to maintain robustness. Although the proposed dual update strategy and classification-regression interaction loss help improve adaptability to target variations and enhance bounding box prediction, their effectiveness remains limited under such extreme conditions. We attribute this to the reliance on multi-template appearance matching, which makes the tracker susceptible to background confusion and motion blur. A more robust solution may require incorporating stronger temporal modeling. Furthermore, as indicated in Table 6, there is still considerable room to improve the tracking speed of our method.

In future work, we aim to enhance the tracker's robustness in dynamic and cluttered environments, while further boosting inference efficiency. Specifically, we plan to further explore motion-aware mechanisms, temporal context

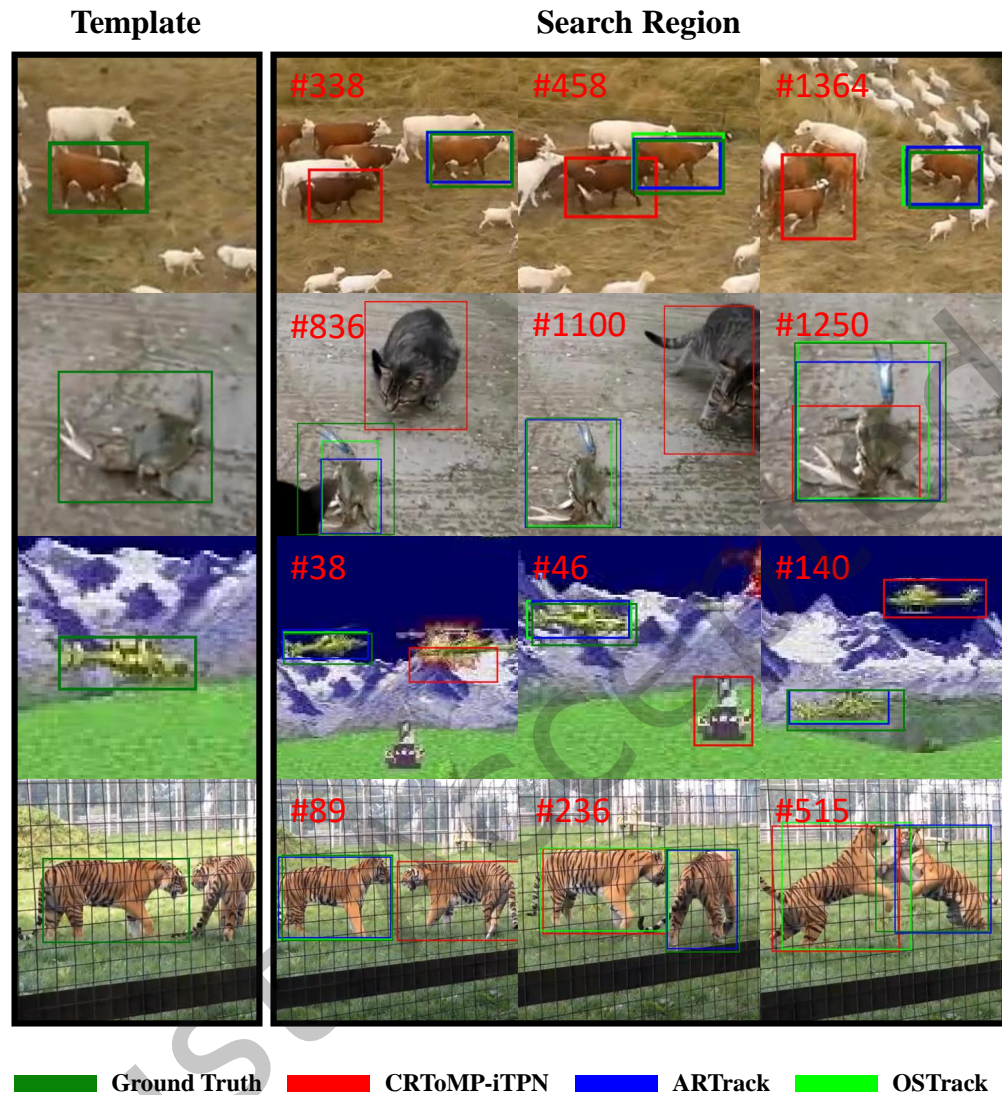


Fig. 6. Visualization failure results of tracking processes. Zoom in for a clearer view.

modeling, and more robust template update strategies, while investigating lightweight tracking frameworks to improve overall tracking speed. For instance, we intend to integrate a Kalman filter to model the target's motion trajectory and use its predictions to refine the tracker's output, thereby achieving more accurate target localization by jointly leveraging appearance and motion information. These efforts are expected to further enhance the stability and generalization capability of the proposed tracker in real-world applications.

5 CONCLUSIONS

In this work, we propose a novel tracker that utilizes dual update strategy and a classification-regression interaction loss for more accurate visual target tracking. The proposed dual update strategy facilitates continuous online template updates by leveraging both image-level and feature-level processes, without compromising the tracker's speed. Meanwhile, the classification-regression interaction loss effectively aligns the classification and regression processes during training, thereby enhancing the accuracy of target bounding box predictions. Collectively, these strategies contribute to a significant improvement in tracking performance, enabling the tracker to achieve a performance level that can compete with the SOTA trackers.

Acknowledgement

This research was supported by the National Natural Science Foundation of China under Grant Nos. 62202362, and 62302073, by the Foundation of Yunnan Key Laboratory of Unmanned Autonomous Systems under Grant No. 202502ZD02, by the China Postdoctoral Science Foundation under Grant Nos. 2022TQ0247 and 2023M742742, by the Guangdong Basic and Applied Basic Research Foundation under Grant Nos. 2024A1515011626, 2025A1515012949, and by the Science and Technology Projects in Guangzhou under Grant No. 2023A04J0397.

References

- [1] Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. 2024. Artrackv2: Prompting autoregressive tracker where to look and how to describe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19048–19057.
- [2] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. 2019. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6182–6191.
- [3] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. 2020. Learning what to learn for video object segmentation. In *European Conference on Computer Vision*. Springer, 777–794.
- [4] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. 2023. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9589–9600.
- [5] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. 2023. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14572–14581.
- [6] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. 2021. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8126–8135.
- [7] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. 2022. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13608–13618.
- [8] Kenan Dai, Yunhua Zhang, Dong Wang, Jianhua Li, Huchuan Lu, and Xiaoyun Yang. 2020. High-performance long-term tracking with meta-updater. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6298–6307.
- [9] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. 2021. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision* 129 (2021), 439–461.
- [10] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. 2019. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5374–5383.
- [11] Zhihong Fu, Zehua Fu, Qingjie Liu, Wenrui Cai, and Yunhong Wang. 2022. SparseTT: Visual Tracking with Sparse Transformers. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 905–912.
- [12] Qi Gao, Mingfeng Yin, Xiang Wu, Di Liu, and Yuming Bo. 2024. Online Multi-Scale Classification and Global Feature Modulation for Robust Visual Tracking. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 7 (2024), 5321–5334.
- [13] Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. 2022. Aiatrack: Attention in attention for transformer visual tracking. In *European Conference on Computer Vision*. Springer, 146–164.
- [14] Shenyuan Gao, Chunluan Zhou, and Jun Zhang. 2023. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18686–18695.
- [15] Jiawei Ge, Jiuxin Cao, Xiangmei Chen, Xuelin Zhu, Weijia Liu, Chang Liu, Kun Wang, and Bo Liu. 2025. Beyond visual cues: Synchronously exploring target-centric semantics for vision-language tracking. *ACM Transactions on Multimedia Computing, Communications and Applications* 21, 5 (2025), 1–21.
- [16] Bing He, Fasheng Wang, Xing Wang, Haojie Li, Fuming Sun, and Hui Zhou. 2024. Temporal context and environment-aware correlation filter for UAV object tracking. *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024), 1–15.

- [17] Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. 2023. Target-aware tracking with long-term context attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 773–780.
- [18] Lianghai Huang, Xin Zhao, and Kaiqi Huang. 2019. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 5 (2019), 1562–1577.
- [19] Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. 2025. Mambalct: Boosting tracking via long-term context state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 4986–4994.
- [20] Yongxin Li, Mengyuan Liu, You Wu, Xucheng Wang, Xiangyang Yang, and Shuiwang Li. 2024. Learning Adaptive and View-Invariant Vision Transformer for Real-Time UAV Tracking. In *Forty-first International Conference on Machine Learning*.
- [21] Donghai Liao, Xiu Shu, Zhihui Li, Qiao Liu, Di Yuan, Xiaojun Chang, and Zhenyu He. 2025. Fine-Grained Feature and Template Reconstruction for TIR Object Tracking. *IEEE Transactions on Circuits and Systems for Video Technology* 35, 9 (2025), 9276–9286.
- [22] Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. 2024. Tracking meets lora: Faster training, larger model, stronger performance. In *European Conference on Computer Vision*. Springer, 300–318.
- [23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer vision*. 740–755.
- [24] Shuai Liu, Dongye Liu, Khan Muhammad, and Weiping Ding. 2021. Effective template update mechanism in visual tracking with background clutter. *Neurocomputing* 458 (2021), 615–625.
- [25] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. 2022. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8731–8740.
- [26] Matthias Mueller, Neil Smith, and Bernard Ghanem. 2016. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*. Springer, 445–461.
- [27] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. 2018. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *European Conference on Computer Vision*. 300–317.
- [28] Hyeonseob Nam and Bohyung Han. 2016. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition*. 4293–4302.
- [29] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. 2022. Robust visual tracking by segmentation. In *European Conference on Computer Vision*. Springer, 571–588.
- [30] Hamid Reza Tofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. 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*. 658–666.
- [31] Jian Shi, Zheng Chang, Yang Yu, Junze Shi, and Haibo Luo. 2025. LATrack: Limited Attention for Visual Object Tracking. *IEEE Access* 13 (2025), 4034–4047.
- [32] Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. 2024. Explicit visual prompts for visual object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4838–4846.
- [33] Zikai Song, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. 2022. Transformer tracking with cyclic shifting window attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8791–8800.
- [34] Fuming Sun, Tingting Zhao, Bing Zhu, Xu Jia, and Fasheng Wang. 2023. Deblurring transformer tracking with conditional cross-attention. *Multimedia Systems* 29, 3 (2023), 1131–1144.
- [35] Zhangyong Tang, Tianyang Xu, Xiao-Jun Wu, and Josef Kittler. 2024. Multi-level fusion for robust RGBT tracking via enhanced thermal representation. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 10 (2024), 1–24.
- [36] Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. 2024. Fast-iTPN: Integrally Pre-Trained Transformer Pyramid Network With Token Migration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 9766–9779.
- [37] Qinghui Wang, Peng Yang, and Lei Dou. 2024. Learning Attention Through Hierarchical Architecture for Visual Object Tracking. *IEEE Signal Processing Letters* 31 (2024), 186–190.
- [38] Shilei Wang, Zhenhua Wang, Qianqian Sun, Gong Cheng, and Jifeng Ning. 2024. Modeling of Multiple Spatial-Temporal Relations for Robust Visual Object Tracking. *IEEE Transactions on Image Processing* 33 (2024), 5073–5085.
- [39] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. 2023. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9697–9706.
- [40] Jingjing Wu, Xi Zhou, Xiaohong Li, Hao Liu, Meibin Qi, and Richang Hong. 2024. Asymmetric Deformable Spatio-temporal Framework for Infrared Object Tracking. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 10 (2024), 1–24.
- [41] Fei Xie, Zhongdao Wang, and Chao Ma. 2024. Diffusiontrack: Point set diffusion model for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19113–19124.
- [42] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. 2021. Learning Spatio-Temporal Transformer for Visual Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10448–10457.

- [43] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*. Springer, 341–357.
- [44] Bin Yu, Ming Tang, Linyu Zheng, Guibo Zhu, Jinqiao Wang, Hao Feng, Xuetao Feng, and Hanqing Lu. 2021. High-performance discriminative tracking with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9856–9865.
- [45] Di Yuan, Xiaojun Chang, Po-Yao Huang, Qiao Liu, and Zhenyu He. 2020. Self-supervised deep correlation tracking. *IEEE Transactions on Image Processing* 30 (2020), 976–985.
- [46] Di Yuan, Xiaojun Chang, Zhihui Li, and Zhenyu He. 2022. Learning adaptive spatial-temporal context-aware correlation filters for UAV tracking. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 3 (2022), 1–18.
- [47] Di Yuan, Xiaojun Chang, Qiao Liu, Yi Yang, Dehua Wang, Minglei Shu, Zhenyu He, and Guangming Shi. 2024. Active Learning for Deep Visual Tracking. *IEEE Transactions on Neural Networks and Learning Systems* 35, 10 (2024), 13284–13296.
- [48] Di Yuan, Xiu Shu, Qiao Liu, and Zhenyu He. 2023. Aligned Spatial-Temporal Memory Network for Thermal Infrared Target Tracking. *IEEE Transactions on Circuits and Systems II: Express Briefs* 70, 3 (2023), 1224–1228.
- [49] Mingshu Zhang, Fangmei Chen, Fasheng Wang, Binbin Wang, Hanwei Li, and Fuming Sun. 2025. Lightweight adaptive multi-scale asymmetric tracker. *Applied Soft Computing* (2025), 114022.
- [50] Zhihao Zhang, Jun Wang, Shengjie Li, Lei Jin, Hao Wu, Jian Zhao, and Bo Zhang. 2024. Review and analysis of rgbt single object tracking methods: A fusion perspective. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 8 (2024), 1–27.
- [51] Zhibin Zhang, Wanli Xue, Qinghua Liu, Kaihua Zhang, and Shengyong Chen. 2023. Learnable diffusion-based amplitude feature augmentation for object tracking in intelligent vehicles. *IEEE Transactions on Intelligent Vehicles* 9, 4 (2023), 4749–4758.
- [52] Zhibin Zhang, Wanli Xue, Yuxi Zhou, Kaihua Zhang, and Shengyong Chen. 2024. Hunt-inspired Transformer for visual object tracking. *Pattern Recognition* 156 (2024), 110703.
- [53] Moju Zhao, Kei Okada, and Masayuki Inaba. 2021. Trtr: Visual tracking with transformer. *arXiv preprint arXiv:2105.03817* (2021).
- [54] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. 2017. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems* 11, 2 (2017), 68–75.
- [55] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. 2024. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 7588–7596.
- [56] Hong Zhu, Qingyang Lu, Lei Xue, Guanglin Yuan, and Kaihua Zhang. 2025. Joint feature extraction and alignment in object tracking with vision-language model. *Engineering Applications of Artificial Intelligence* 152 (2025), 110787.