ODONet: Online Dynamic Offset Network for Visual Object Tracking

Qinghua Liu[1], Wanli Xue[1,2*], and Shengyong Chen[1]

[1] Tianjin University of Technology, Tianjin, China
[2] Qinghai Nationalities University, Xining, China
liuqinghua@stud.tjut.edu.cn, xuewanli@email.tjut.edu.cn, sy@ieee.org

Abstract. Offset learning has recently demonstrated remarkable prowess in capturing the structural diversity of objects. Consequently, it appears to be a natural fit for modeling the discriminative features of visual targets in tracking. Yet existing leading offline offset-based methods struggle to cope with the ever-changing appearance and motion of the target during tracking. To address this limitation, we present ODONet, a Transformer-based tracker that performs online dynamic offset network. Its Dynamic Visual–Motion Feature Interaction (DV-MFI) method fuses target appearance with historical box information, yielding appearance features and motion-aware embeddings. These are then fed to an Online Dynamic Offset Propagation (ODOP) method that explicitly captures inter-frame motion trends, augments them with offline offsets, and steers deformable attention toward an adaptive receptive field. To our knowledge, ODONet is the first tracker to propagate dynamic offsets online, with excellent performance on publicly available datasets, demonstrating its robustness to appearance and motion drift. Code, models, and additional information are available at https://github.com/WhiteButterflies/ODONet.

Keywords: Visual Object Tracking · Online Dynamic Offset Learning · Deformable Attention

1 Introduction

Visual object tracking aims to accurately locate and follow a target in a video sequence. To better model the appearance and motion of the target, recent works incorporate additional visual features and target's historical bounding boxes into the tracking pipeline. Representative approaches include expanding the region of interest (ROI) and employing autoregressive sequence modeling. As shown in Fig. 1(A1), trackers [17,21,30,36,51] adopt full-frame search strategies to enlarge the receptive field, mitigating potential target disappearance. Other methods exploit spatiotemporal information by concatenating features [13, 41, 42], propagating hidden states [22, 47], or using memory structures such as PN tree [19] to

* Wanli Xue is the corresponding author (xuewanli@email.tjut.edu.cn).

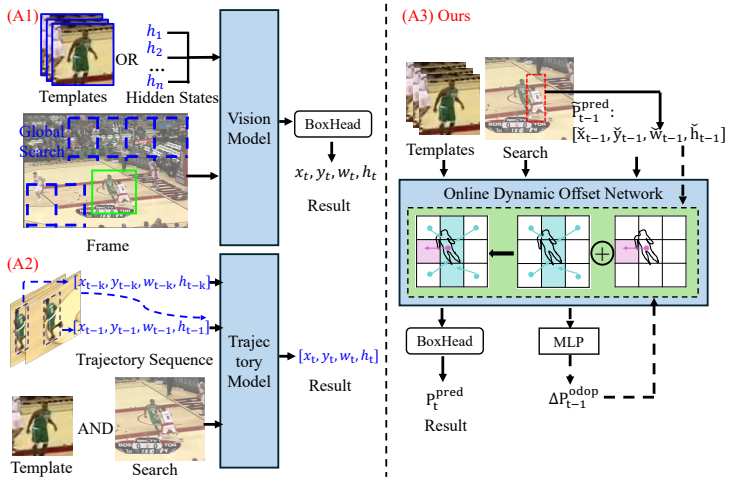

Fig. 1: Different types of trackers: (A1) receptive field expansion (A2) autoregressive trajectory modeling, and (A3) our proposed online dynamic offset network.

extend temporal context. However, expanding the receptive field inevitably introduces irrelevant background and distractor objects during feature extraction and fusion, increasing the risk of localization errors.

To address temporal modeling, as shown in Fig. 1(A2), another trackers [7,38] reformulate tracking as an autoregressive sequence generation problem, where the current target state is predicted based on historical bounding boxes. Yet, these approaches typically integrate historical boxes only at the decoding stage, neglecting the fine-grained interactions between coordinate semantics and visual features during encoding. Whether through spatial expansion or sequence modeling, most existing methods still rely on indirect motion inference and lack explicit modeling of inter-frame displacement.

Recently, in the field of object detection, offline offset-based methods such as deformable convolution [9] and deformable attention [40] have been proposed to better adapt to diverse object structures. However, these offsets are generated for different objects across categories, without fully leveraging the spatiotemporal continuity of the same target in tracking scenarios. This raises a natural question: Can we directly model the inter-frame displacement of the target and actively guide the network to focus on the most informative regions?

We propose ODONet, as shown in Fig. 1(A3), a Transformer-based tracking framework, that answers this question affirmatively. For explicit displacement modeling, we draw inspiration from SAM [23] prompt-based interaction mechanism and design a **Dynamic Visual-Motion Feature Interaction (DV-MFI)** method, where historical bounding boxes are treated as motion prompts. DV-MFI takes both visual features and motion prompts as input and leverages a Two-Way Transformer (V&M Interaction Encoder) to produce motion trend

encodings enriched with appearance cues and visual features conditioned on motion dynamics.

To further adapt to dynamic scenarios, we introduce an **Online Dynamic Offset Propagation (ODOP)** method to replace traditional offline offset generation. In each inference step, ODOP converts the motion trend encoding from the previous frame into online offsets, which are then injected into the offline offsets predicted from the current frame visual features. This results in a new form of online dynamic offsets, enabling offset generation to be continuous and history-aware, rather than restricted to the current frame. These offsets directly guide feature sampling in the search area, dynamically adjusting the receptive field according to the target's motion trajectory. In summary, the contributions are as follows:

1. Inspired by SAM, our DV-MFI method integrates target appearance with historical bounding-box information to generate dynamic target offsets that can be propagated online during inference.
2. While most offset methods are unconstrained offline approaches, the proposed ODOP method injects historical online target offsets into the current offline offsets during testing. This results in online dynamic offsets that better adapt to dynamic scenes involving the same target.
3. We propose the first online dynamic offset network in visual tracking, which learns object offsets during training and propagates online dynamic offsets to adjust the receptive field during testing. The corresponding visualization scheme serves as a dynamic perspective for analyzing the intrinsic working mechanisms of the tracker.

2 Related Work

2.1 Offset Learning in Detection and Segmentation

Offset learning introduces spatial sampling offsets to enhance the network ability to handle object deformation, scale variation, and complex spatial structures. In object detection, deformable convolution was first proposed in DCN [9], where learnable offsets are added to standard convolution operations. By forming spatially adaptive kernels, this mechanism enables dynamic perception of geometric structure. DCNv2 [49] further extended this idea by incorporating more deformable convolutions and assigning importance weights to offset sampling points, thus improving the focus on target regions.

Deformable attention generalizes the offset mechanism to Transformer architectures. DeformDETR [50] applied deformable attention in the DETR [5] framework, using offsets around reference points to reduce the attention's spatial range and transform its complexity from quadratic to linear. In semantic segmentation, DefVis [44] introduced deformable attention into video instance segmentation to enhance training efficiency by focusing on important spatiotemporal positions. Cai [3] proposed a Dynamic Sparse Cross-modality Fusion module that generates modality-specific deformable points from RGB and X-modality offset networks, enabling efficient and dynamic cross-modal fusion.

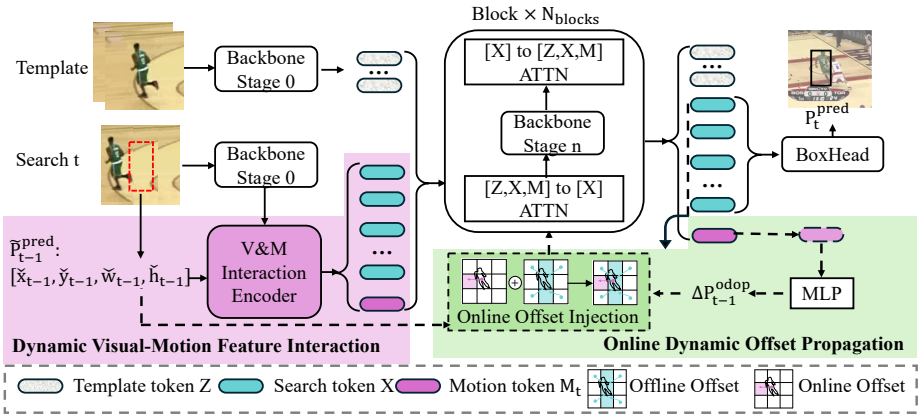

Fig. 2: Framework of the proposed ODONet.

2.2 Offset Learning in Visual Tracking

In the tracking domain, tracker [39] applied deformable convolution to the RPN module, using offset range constraints and modulation scalars to address CNN limitations in rotation and scale adaptation. SiamDCN [46] introduced deformable cross-correlation into the SiamFC [2], enabling dynamic adjustment of correspondence between template and search area features to cope with significant object deformations. D-TransT [48] extended offset sampling points to multi-scale feature fusion, resulting in faster convergence and better prediction accuracy. Other works [18,33] employed deformable attention modules to adapt spatial sampling patterns, refining the feature extraction region for more comprehensive representation.

In summary, whether used in detection or tracking, most existing offset learning methods generate offsets offline within a single inference step. These methods typically constrain only the range—not the values—of offsets during training, aiming to increase the discriminative power of sampling points across different object types. However, such approaches break the temporal continuity of object motion. In contrast, ODONet focuses on maintaining appearance and motion consistency across frames for the same target, addressing the limitations of prior offline offset methods.

3 Method

This section presents our proposed ODONet in detail. We first introduce the overall framework, followed by the description of the Dynamic Visual-Motion Feature Interaction (DV-MFI) method, the Online Dynamic Offset Propagation (ODOP) method and motion constraints with loss function.

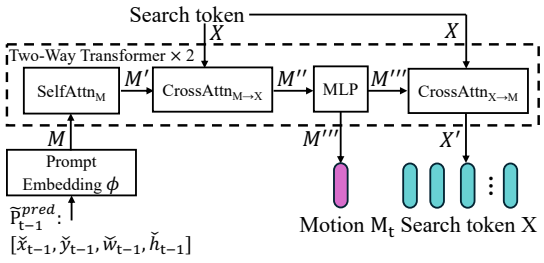

Fig. 3: The details of Dynamic Visual-Motion Feature Interaction (DV-MFI).

3.1 Overview

As shown in Fig. 2, the proposed ODONet tracking framework consists of a multi-stage Transformer backbone, a Dynamic Visual-Motion Feature Interaction (DV-MFI) method, an Online Dynamic Offset Propagation (ODOP) method, multiple interaction blocks, and a bounding box prediction head. The input includes N template images, the current search area, and the previous frame target bounding box.

First, both templates and search area images are passed through Stage-0 of the backbone for patch embedding, yielding initial visual features $Z \in \mathbb{R}^{(N\times\frac{H_Z}{4^2}\times\frac{W_Z}{4^2})\times C_0}$ and $X \in \mathbb{R}^{(1\times\frac{H_X}{4^2}\times\frac{W_X}{4^2})\times C_0}$, where H_Z and W_Z are the image dimensions and 4 denotes the downsampling ratio.

Subsequently, X interacts with the previous frame bounding box $\widetilde{P}_{t-1}^{pred}$ through the DV-MFI method to obtain tokens encoding both appearance and motion trends. These tokens are passed into a stack of Transformer blocks that perform cross-attention, backbone feature extraction on all tokens, and another round of cross-attention, enabling deep fusion of visual and motion information.

During this process, the ODOP method combines historical motion cues with current features to generate online and offline offsets. Using the Online Offset Injection (OOI) strategy, these offsets guide deformable feature sampling to dynamically adjust the receptive field. Finally, the prediction head regresses the current frame target bounding box and offset vector.

3.2 Dynamic Visual-Motion Feature Interaction

Unlike existing methods that model historical boxes as global-coordinate sequences—which often causes misalignment during local search—we normalize the previous bounding box relative to the current search area to obtain $\widetilde{P}_{t-1}^{pred} = [\tilde{x}_{t-1}, \tilde{y}_{t-1}, \tilde{w}_{t-1}, \tilde{h}_{t-1}] \in [0,1]^4$. Note that while the search area is cropped based on the previous target center (Fig. 2), it may not fully enclose the moving object. Inspired by SAM [23], we treat this geometric prior as a motion prompt and embed it into a sparse motion representation via a prompt encoding function:

$$M = \phi(\widetilde{P}_{t-1}^{pred}), \tag{1}$$

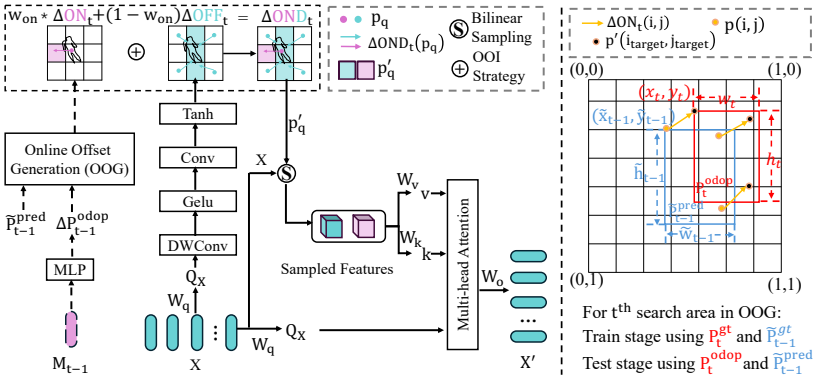

Fig. 4: Left: The details of Online Dynamic Offset Propagation (ODOP) method. Right: Illustration of the affine transformation in Online Offset Generation (OOG) module.

where M is a sparse motion trend encoding vector.

To enable early cross-modal interaction, M and the backbone Stage-0 search features X are fed into a Two-Way Transformer [23]. As shown in Fig. 3, rather than delaying interaction to the decoding stage, this module performs sequential operations: self-attention on M, cross-attention from M to X, an MLP update, and reverse cross-attention from X to M. This yields the updated representation pair (X, M_t), directly injecting temporal motion priors into low-level spatial features during the encoding phase to enhance adaptability in dynamic tracking scenarios.

3.3 Online Dynamic Offset Propagation

To enable the tracker to actively adapt to continuous target motion rather than passively reacting to it, we propose an Online Dynamic Offset Propagation (ODOP) method. The key idea is that, for each tracking frame, we generate two complementary offset fields: (1) an online offset field ΔON_t, which is predicted from historical motion and provides a strong motion prior (e.g., translation and scale change) for deformable feature sampling; and (2) an offline offset field ΔOFF_t, which is learned from current search area features to capture the global motion of the receptive field.

We then fuse these two offsets to obtain the final Online Dynamic Offset ΔOND_t, which guides the deformable attention module to perform motion-aware feature aggregation. In this way, the receptive field is no longer globally fixed as in standard vanilla ViT [10], but is dynamically guided by the historical motion state, enabling faster and more accurate localization of rapidly moving or drastically deforming targets.

The whole process is illustrated in Fig. 4 left part, and is detailed in the three steps.

Online Offset Generation. Online Offset Generation (OOG) module is constrained by the rectangular bounding-box annotations: within a short time range, the relative positions of points inside the bounding box remain unchanged. Based on this assumption, we compute a offset field ΔON_t for each feature point inside the bounding box at frame t–1, pointing to its predicted location at frame t.

We first utilize the motion trend encoding M_{t-1}, and predict the bounding box offset $\Delta P_{t-1}^{odop} \in [-1,1]^4$ from frame $t-2$ to $t-1$ through a lightweight prediction head (MLP), which represents the translation and scale of the object bounding box:

$$\Delta P_{t-1}^{odop} = \mathrm{MLP}(M_{t-1}). \tag{2}$$

Given the bounding box $\widetilde{P}_{t-1}^{pred}$ in the search area of frame t and the offset ΔP_{t-1}^{odop}, we obtain the ODOP motion-predicted bounding box by addition:

$$P_t^{odop} = \widetilde{P}_{t-1}^{pred} + \Delta P_{t-1}^{odop}. \tag{3}$$

For any feature-grid point $p(i,j)$ inside $\widetilde{P}_{t-1}^{pred}$, its corresponding location $p'(i_{\text{target}}, j_{\text{target}})$ inside P_t^{odop} can be computed by an affine transformation [20], as illustrated in Fig. 4 right part.

For the point $p(i,j)$ inside $\widetilde{P}_{t-1}^{pred}$, we compute its relative coordinates:

$$r_x = \frac{i - \tilde{x}_{t-1}}{\tilde{w}_{t-1}}, \qquad r_y = \frac{j - \tilde{y}_{t-1}}{\tilde{h}_{t-1}}. \tag{4}$$

According to the object deformation, it is mapped into P_t^{odop}:

$$p' = (x_t + r_x w_t,\ y_t + r_y h_t). \tag{5}$$

Substituting Eq. 4 into Eq. 5, the ratios $\frac{w_t}{\tilde{w}_{t-1}}$ and $\frac{h_t}{\tilde{h}_{t-1}}$ represent the horizontal and vertical scale changes of the target, respectively.

Thus, the displacement vector of each feature point $p(i,j)$ inside $\widetilde{P}_{t-1}^{pred}$ is:

$$\Delta ON_t(i,j) = p' - p = (i_{\text{target}} - i,\ j_{\text{target}} - j). \tag{6}$$

Finally, by concatenating all feature points inside $\widetilde{P}_{t-1}^{pred}$, we construct a dense online motion prior:

$$\Delta ON_t \in \mathbb{R}^{\frac{H_X}{4^2} \times \frac{W_X}{4^2} \times 2},$$

and the offsets of feature points outside $\widetilde{P}_{t-1}^{pred}$ are filled with zero.

Offline Offset Prediction. In parallel with Online Offset Generation, we predict an offline offset field ΔOFF_t from the search area feature X at frame t, which has been enriched with historical bounding box information. Its purpose is to capture the global motion behavior of the receptive field.

Specifically, we perform a linear projection on the search area feature X and generate the offline offset field by applying Eq. 7:

$$\Delta OFF_t = \tanh\big(\mathrm{Conv}(\mathrm{GELU}(\mathrm{DWConv}(X \cdot W_q)))\big) \tag{7}$$

where W_q performs a linear mapping on the visual feature X to obtain the query Q_X, and the $\tanh(\cdot)$ activation is used to constrain the offset within the receptive field.

Online-Guided Feature Sampling. Given the online offset field ΔON_t, which represents the motion prior, and the offline offset field ΔOFF_t, which represents the receptive field behavior, we fuse them through the Online Offset Injection (OOI) strategy in Eq. 8 to incorporate both historical and current motion cues. The resulting online dynamic offset ΔOND_t is used to guide feature sampling as formulated in Eq. 9:

$$\Delta OND_t = w_{on} \cdot \Delta ON_t + (1 - w_{on}) \cdot \Delta OFF_t, \tag{8}$$

where w_{on} is a hyper-parameter controlling the contribution of the online offset field.

Finally, the fused ΔOND_t directly guides the deformable attention module during feature sampling. Specifically, instead of sampling features on a fixed regular grid as in vanilla ViT [10], each query point p_q dynamically determines its sampling location based on the motion cue from ΔOND_t. Given one sampling point p_q of the original sampling grid in deformable attention [40], the motion-guided sampling point is:

$$p'_q = p_q + \Delta OND_t(p_q). \tag{9}$$

If the sampling point p'_q falls outside the feature map boundary, it is clipped to the nearest valid location within the feature map. The feature at position p'_q is bilinearly interpolated from the search feature map X, and is further projected by W_k and W_v to form the sampled key k and value v. The final output of the deformable attention is:

$$X' = \mathrm{MultiHeadAttn}(Q_X, k, v)W_o. \tag{10}$$

Through Eqs. 8–10, the historical motion information is effectively injected into the current-frame feature extraction pipeline. As shown in Fig. 5(b), the updated sampling points p'_q demonstrate dynamically adjusted receptive fields, in contrast to vanilla ViT.

3.4 Motion Constraints and Loss Function

To ensure that the predicted bounding box motion ΔP_t^{odop} and the predicted offline offset field ΔOFF_t are consistent with the tracking objective, we design an end-to-end loss function that jointly constrains different prediction outputs in ODOP. The supervised targets mainly include three aspects: overall motion vector prediction, offline offset prediction, and final bounding box regression.

Overall Motion Vector Supervision. As formulated in Eq. 11, the motion trend encoding M_{t-1} predicts a four-dimensional vector ΔP_t^{odop} that represents the global motion of the target. To ensure accurate estimation, we compare this

prediction with the ground-truth motion vector ΔP_t^{gt} between two frames and apply an L1 loss:

$$\Delta P_t^{gt} = P_t^{gt} - \widetilde{P}_{t-1}^{gt}, \tag{11}$$

$$L_{\text{motion}} = \left\| \Delta P_t^{odop} - \Delta P_t^{gt} \right\|_1. \tag{12}$$

Here, P_t^{gt} denotes the ground-truth bounding box at frame t, and $\widetilde{P}_{t-1}^{gt}$ denotes the ground-truth box at frame $t-1$ re-normalized into the coordinate system of frame t. The boxes notations are shown in Fig 4 right part. The motion loss L_{motion} encourages the model to learn accurate translation and scale changes of the target.

Offline Offset Supervision. Since the offline offset field ΔOFF_t is designed to capture the global motion behavior of the receptive field, a dense pixel-level supervision signal is required. Ideally, the offline offset should precisely describe the pixel-wise mapping from $\widetilde{P}_{t-1}^{gt}$ to P_t^{gt}. Therefore, we use the online offset generation process described in Sec. 3.3 to construct a pseudo ground-truth offline offset field:

$$\Delta OFF_t^{gt} = \text{OnlineOffsetGeneration}(\widetilde{P}_{t-1}^{gt}, P_t^{gt}), \tag{13}$$

where Online Offset Generation module refers to the complete process defined by Eq. 3 - 5. We then apply both L1 loss and cosine similarity loss to constrain the predicted ΔOFF_t. The L1 loss enforces the correct magnitude of the motion, while the cosine loss enforces its direction consistency:

$$\mathcal{L}_{\text{offset-l1}} = \left\| \Delta OFF_t - \Delta OFF_t^{gt} \right\|_1, \tag{14}$$

$$\mathcal{L}_{\text{offset-cos}} = 1 - \frac{\left\langle \Delta OFF_t, \Delta OFF_t^{gt} \right\rangle}{\|\Delta OFF_t\| \left\|\Delta OFF_t^{gt}\right\|}. \tag{15}$$

It is worth noting that Eq. 14 and Eq. 15 are only computed within the region covered by $\widetilde{P}_{t-1}^{gt}$ inside the search area at frame t.

Bounding Box Regression and Total Loss. For classification and box prediction, we adopt the OSTrack [45] head. A focal loss $\mathcal{L}_{\text{focal}}$ is used for classification, while the predicted bounding box is supervised by both L1 loss $\mathcal{L}_{\text{box-l1}}$ and GIoU loss $\mathcal{L}_{\text{giou}}$.

The total loss is a weighted sum of all components:

$$\begin{aligned} \mathcal{L} = \; & \lambda_f \mathcal{L}_{\text{focal}} + \lambda_b \mathcal{L}_{\text{box-l1}} + \lambda_g \mathcal{L}_{\text{giou}} \\ & + \lambda_m \mathcal{L}_{\text{motion}} + \lambda_o \mathcal{L}_{\text{offset-l1}} + \lambda_c \mathcal{L}_{\text{offset-cos}}, \end{aligned} \tag{16}$$

where we set $\lambda_b = 5$, $\lambda_g = 2$, and all other weights to 1. This joint loss formulation allows the model to be trained end-to-end, encouraging ODONet to maintain robustness and accuracy in dynamic tracking scenarios.

Table 1: Comparison with state-of-the-arts on four popular benchmarks: LaSOT, $\text{LaSOT}_{\text{ext}}$, GOT-10K, and TrackingNet. Where * denotes for trackers only trained on GOT-10k. Best in bold, second best underlined.

Method	LaSOT			LaSOT_{ext}			TrackingNet			GOT-10k*		
	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$SR_{0.5}$	$SR_{0.75}$
ODONet-B224	74.6	<u>84.3</u>	<u>82.3</u>	**54.3**	**65.5**	**61.8**	86.3	<u>90.8</u>	86.3	78.9	<u>89.6</u>	77.9
ODONet-B384	**75.1**	**84.5**	**82.7**	53.1	63.7	59.7	**87.4**	**91.6**	**88.2**	**81.4**	**90.0**	**81.1**
SPMTrack-B378 [4]	<u>74.9</u>	84.0	81.7	-	-	-	86.1	90.2	85.6	76.5	85.9	76.3
SUTrack-B384 [6]	74.4	83.9	81.9	52.9	63.6	60.1	<u>86.5</u>	90.7	<u>86.8</u>	79.3	88.0	<u>80.0</u>
DreamTrack-B256 [15]	73.8	83.4	80.6	53.1	64.1	59.8	85.8	90.0	85.3	77.5	87.1	74.2
SUTrack-B224 [6]	73.2	83.4	80.5	53.1	64.2	60.5	85.7	90.3	85.1	77.9	87.5	78.5
ARPTrack-B256 [26]	72.6	81.4	78.5	52.0	62.9	58.7	85.5	90.0	85.3	77.7	87.3	74.3
MambaLCT-256 [25]	71.8	83.0	79.4	51.6	64.0	59.0	84.3	89.2	83.9	74.8	85.4	72.1
ARTrackV2-L384 [1]	73.6	82.8	81.1	<u>53.4</u>	63.7	60.2	86.1	90.4	86.2	<u>79.5</u>	87.8	79.6
ODTrack-B384 [47]	73.2	83.2	80.6	52.4	63.9	60.1	85.1	90.1	84.9	77.0	87.9	75.1
LoRAT-B378 [27]	72.9	81.9	79.1	53.1	<u>64.8</u>	<u>60.6</u>	84.2	88.4	83.0	73.7	82.6	72.9
ARTrackV2-256 [1]	71.6	80.2	77.2	50.8	61.9	57.7	84.9	89.3	84.5	75.9	85.4	72.7
EVPTrack-224 [35]	70.4	80.9	77.2	48.7	59.5	55.1	83.5	88.3	-	73.3	83.6	70.7
ARTrack-L384 [38]	73.1	82.2	80.3	52.8	62.9	59.7	85.6	89.6	86.0	78.5	87.4	77.8
SeqTrack-L384 [7]	72.5	81.5	79.3	50.7	61.6	57.5	85.5	89.8	85.8	74.8	81.9	72.2
ARTrack-256 [38]	70.4	79.5	76.6	46.4	56.5	52.3	84.2	88.7	83.5	73.5	82.2	70.9
Mixformer-22k [8]	69.2	78.7	74.7	-	-	-	83.1	88.1	81.6	70.7	80.0	67.8
OSTrack-256 [45]	69.1	78.7	75.2	47.4	57.3	53.3	83.1	87.8	82.0	71.0	80.4	68.2

4 Experiments

4.1 Implementation Details

Model. ODONet is implemented in Python 3.10 and PyTorch 2.7.1. All experiments are conducted on an Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz with 78.3GB of memory and a single 2080Ti GPU. We adopt Fast-iTPN-B as the backbone model and use 5 templates. For ODONet-B224, the search-region resolution is set to 224×224 and the template resolution is 112×112. For ODONet-B384, the search-region resolution is 384×384 and the template resolution is 192×192. For deformable attention, we use 8 attention heads, and the sampling points p_q correspond to all spatial locations of the search area feature map X. Unlike DeformDETR [50], our queries Q_X share the same offset field. If the sampling point p'_q falls outside the feature-map boundary, it is clipped to the boundary.

Training. The training set comprises LaSOT [12], TrackingNet [32], GOT-10k [16], COCO [28], and VastTrack [34]. For GOT-10k evaluation, only its training subset is used, in accordance with the official protocol. We use AdamW [29] as the optimizer, setting the learning rate to 4×10^{-5} for the backbone and 4×10^{-4} for other modules, with a weight decay of 10^{-4}. In each training epoch, 60K image pairs are sampled uniformly from the dataset pool. Training is conducted for 300 epochs (100 for GOT-10k), with a learning rate decay at epoch 240 (80 for GOT-10k). The model is trained on 4 RTX 3090 GPUs with a total batch size of 128.

4.2 Results and Comparisons

Table 2: Comparison with state-of-the-art methods on TNL2K, UAV123 and VOT2020 benchmarks in score.

	SiamFC	MDNet	Ocean	TransT	ATOM	STARK	MixFormer	OSTrack	SeqTrack	MambaLCT	ODONet-B224	ODONet-B384
TNL2K(AUC)	29.5	38.0	38.4	50.7	44.7	-	-	55.9	56.4	58.5	**62.4**	**63.1**
UAV123(AUC)	46.8	52.8	57.4	68.1	64.3	68.2	69.5	68.3	68.6	70.1	**71.1**	**71.4**
VOT2020(EAO)	-	-	43.0	-	27.1	49.7	55.5	52.4	56.1	-	**61.6**	**61.5**

We compare our ODONet with state-of-the-art trackers on Tab. 1-2. Evaluation is performed on large-scale benchmarks (LaSOT [12], LaSOT$_{ext}$ [11], TrackingNet [32], GOT-10k [16]) and smaller ones (VOT2020 [24],TNL2K [37], UAV123 [31]).

LaSOT LaSOT is a large-scale long-term visual object tracking benchmark, containing 3.52 million frames with an average sequence length of 2,512 frames, significantly exceeding other datasets of its kind. It consists of 70 object categories, each with 20 different instances. Our method achieves an AUC score of 75.1%, outperforming all the compared trackers. Although our ODONet-B224 obtains an AUC of 74.6%, which is slightly lower than SPMTrack-B378 (74.9%), our model uses a smaller input resolution while achieving better P$_{norm}$ and P metrics.

LaSOT$_{ext}$ This extension to LaSOT adds language annotations, 15 new categories, and 150 more videos. Our tracker achieves 54.3% AUC.

TrackingNet TrackingNet contains over 30K videos with 114 hours of footage. The test set is evaluated via an online server. Our tracker achieves 87.4% AUC and 91.6% P$_{norm}$, outperforming all listed trackers, indicating strong center localization performance.

GOT-10k GOT-10k includes 10K+ videos, 560+ classes, and 87 motion types, designed for diverse motion modeling. Following the one-shot protocol, we train only on its training split, achieve 81.4% AO, outperforming all listed trackers. As GOT-10k enforces non-overlapping object classes between training and testing, our result suggests stronger motion-aware modeling beyond appearance matching.

VOT2020, TNL2K and UAV123 VOT2020 features appearance changes, occlusions, and clutter. We apply Alpha-Refine [43] for pixel-level refinement, achieve 61.6 EAO. TNL2K, a text-guided tracking benchmark, we achieve 63.1% AUC. UAV123, which features drone-view videos, we reach 71.4% AUC.

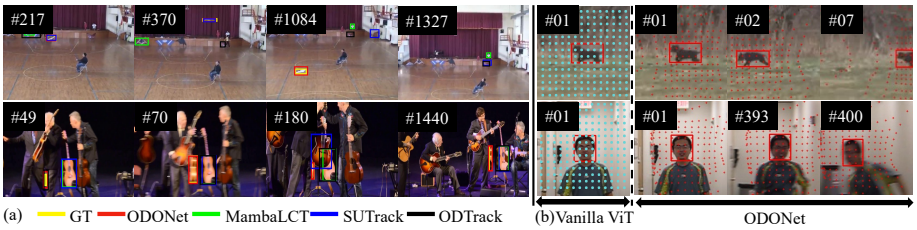

Fig. 5: Left (a) :Visualization on LaSOT fast motion challenge. Zooming in is suggested to compare the tracking results. Right (b): The visualization of our offset field under target fast motion.

4.3 Exploration Studies

To evaluate the contributions of different components, we conduct detailed ablation studies on the GOT-10k dataset.

Table 3: Ablation studies on the GOT-10K benchmark.

(a) Study on method				(b) Online Ratio	
Method	AO	Params(M)	MACs(G)	w_{on}	AO
DV-MFI + ODOP	78.9	90.8	37.7	0.25	78.2
+ODOP	78.0	82.4	37.3	0.50	78.9
+DV-MFI	77.6	89.7	37.5	0.75	77.9
Baseline	77.1	81.3	37.1	1.00	78.5

Study on Our Method. Our ODONet is composed of the proposed DV-MFI method and ODOP method. In Tab. 3(a), the Baseline does not use the DV-MFI method nor the ODOP method, meaning that no historical bounding box information is leveraged and the receptive fields remain fixed. It achieves an AO of 77.1%.

In the variant +**DV-MFI**, we only inject historical bounding box information into the visual features, achieving an AO of 77.6%. This indicates that incorporating historical box information benefits target localization. In the variant +**ODOP**, we remove DV-MFI and only replace the fixed global receptive field of vanilla ViT with deformable attention, achieving an AO of 78.0%. This demonstrates that adaptive receptive field adjustment also improves localization. The variant **DV-MFI**+**ODOP**, which is our full ODONet, predicts object motion based on historical bounding boxes and achieves an AO of 78.9%. This confirms that the combination of historical motion cues and dynamic receptive field adjustment is highly complementary.

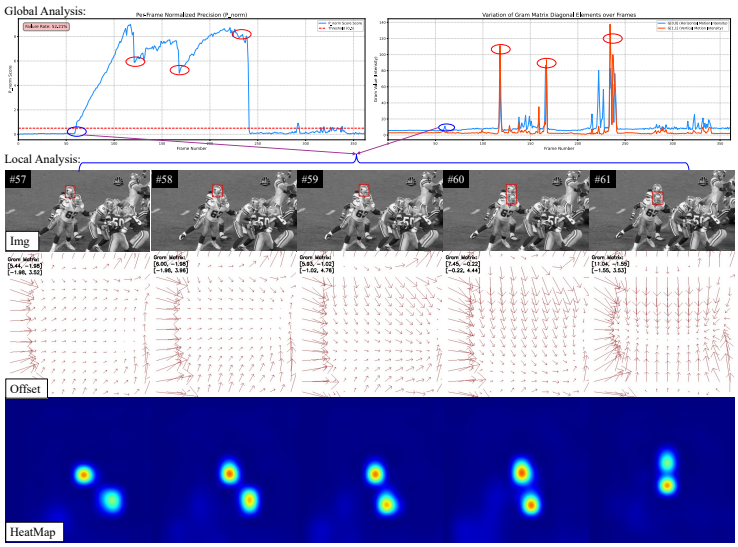

Fig. 6: Visual diagnosis of tracking behavior. **Top (Global Analysis):** When tracking becomes unstable (indicated by P_{norm} over the threshold), our offset statistics (Gram Matrix) exhibit distinct sharp peaks, serving as a clear indicator of tracking failure. **Bottom (Local Analysis):** In drift scenarios (e.g., frame #60), unlike HeatMaps which remain ambiguous, our Offset field **explicitly points towards the distractor before the failure occurs**, providing an early warning of the drift direction.

Compared with the Baseline, ODONet introduces only a 11.7% increase in parameters and a 1.6% increase in Macs, while bringing a substantial AO improvement of 1.8%.

Study on Online Ratio. Our ODOP method fuses ΔON_t and ΔOFF_t through the OOI strategy. In Tab. 3(b), we vary the online offset ratio w_{on}. When $w_{\text{on}} < 0.5$, the AO score increases as w_{on} rises. The best performance is achieved at $w_{\text{on}} = 0.5$, yielding 78.9% AO on GOT-10k. However, increasing w_{on} beyond 0.5 causes performance fluctuations. We attribute this to over-reliance on historical predictions—if the prediction from the previous frame is inaccurate, the error may be amplified and propagated. Setting $w_{\text{on}} = 0.5$ thus strikes a good balance between leveraging historical motion and current observations.

4.4 Visualization and Visual Diagnosis Tool for Tracking Behavior

Visualization. Fig. 5(a) illustrates the tracking results on *kite-10* and *guitar-10*, where our method is able to accurately localize the target despite rapid object movements and frequent occlusions. Fig. 5(b) illustrates the receptive field guided by our dynamic online offset field (Eq. 9). Compared with the global search-region receptive field of the vanilla ViT, our sampling points are more concentrated around the target area.

Visual Diagnosis of Tracking Behavior. Compared with static heatmaps, our offset-based visualization provides a dynamic analysis of the tracker's *feature selection intention*. In Fig. 6 (Bottom), we illustrate a target drift scenario. While the HeatMap (e.g., in OSTrack [45]) only exhibits ambiguous multi-peak responses when the distractor enters the frame (#59-#60), our $\Delta ONDt$ offset field explicitly points toward the distractor one frame earlier (#59), revealing a drift tendency before the bounding box actually shifts. This demonstrates that offsets capture motion-scale dynamics and hesitation earlier than confidence-based heatmaps. **We will release this visualization tool.**

Furthermore, our offset field can evaluate long-term tracking stability without ground-truth supervision. Inspired by Image Style Transfer [14], we compute the Gram matrix of $\Delta ONDt$ to capture the "style" (intensity) of horizontal ($G[0,0]$) and vertical ($G[1,1]$) motions. As shown in Fig. 6 (Top), abnormal peaks in the Gram matrix consistently align with severe performance drops in the frame-wise P_{norm} [32] metric (e.g., during the drift at #57-#61 and subsequent unstable periods). These novel peaks, which deviate from prior motion patterns, serve as strong unsupervised indicators of tracking failure, offering a new reference for dynamic strategy adjustment during inference.

5 Conclusion

We proposed ODONet, an online dynamic offset propagation network for visual tracking that departs from static offset paradigms and emphasizes motion modeling. The DV-MFI method integrates target appearance and historical boxes to produce visual and motion trend encodings. The ODOP method generates offline offsets from visual features and online offsets from motion encodings. These are fused using the OOI strategy to generate online dynamic offsets, which guide the deformable attention for dynamic receptive field adjustment. Our results demonstrate that integrating explicit, continuous motion priors into the feature sampling process is a highly effective strategy to enhance tracking robustness in dynamic scenarios. Our offset-visualization tool further offers a novel perspective for analyzing the behavioral dynamics of the tracker. This opens a new direction for tracker design: from passively responding to object changes to actively predicting and adapting to object motion. We hope this work encourages the community to explore temporal dynamic modeling beyond deep backbones and stronger appearance matching.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62376197, Grant 62020106004, Grant 92048301; and in part by the Tianjin Science and Technology Program under Grant 23JCYBJC00360; and in part by the Key Research and Development Program of Shaanxi Province under Grant 2025CY-YBXM-513; and in part by the Tianjin Health Research Project under Grant TJWJ2025MS044.

References

1. Bai, Y., Zhao, Z., Gong, Y., Wei, X.: Artrackv2: Prompting autoregressive tracker where to look and how to describe. In: CVPR. pp. 19048–19057 (2024)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.S.: Fully-convolutional siamese networks for object tracking. In: ECCV. pp. 850–865 (2016)
3. Cai, J., Su, J., Li, Q., Yang, W., Wang, S., Zhao, T., He, S., Liu, W.: Keep the balance: A parameter-efficient symmetrical framework for rgb+ x semantic segmentation. In: CVPR. pp. 10587–10598 (2025)
4. Cai, W., Liu, Q., Wang, Y.: Spmtrack: Spatio-temporal parameter-efficient fine-tuning with mixture of experts for scalable visual tracking. In: CVPR. pp. 16871–16881 (2025)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229 (2020)
6. Chen, X., Kang, B., Geng, W., Zhu, J., Liu, Y., Wang, D., Lu, H.: Sutrack: Towards simple and unified single object tracking. In: AAAI. vol. 39, pp. 2239–2247 (2025)
7. Chen, X., Peng, H., Wang, D., Lu, H., Hu, H.: Seqtrack: Sequence to sequence learning for visual object tracking. In: CVPR. pp. 14572–14581 (2023)
8. Cui, Y., Jiang, C., Wang, L., Wu, G.: Mixformer: End-to-end tracking with iterative mixed attention. In: CVPR. pp. 13608–13618 (2022)
9. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR. pp. 1–9 (2020)
11. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Huang, M., Liu, J., Xu, Y., et al.: Lasot: A high-quality large-scale single object tracking benchmark. IJCV pp. 439–461 (2021)
12. 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: CVPR. pp. 5374–5383 (2019)
13. Gao, S., Zhou, C., Ma, C., Wang, X., Yuan, J.: AiATrack: Attention in attention for transformer visual tracking. In: ECCV. pp. 146–164 (2022)
14. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
15. Guo, M., Tan, W., Ran, W., Jing, L., Zhang, Z.: Dreamtrack: Dreaming the future for multimodal visual object tracking. In: CVPR. pp. 7201–7210 (2025)
16. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE TPAMI pp. 1562–1577 (2019)

17. Huang, L., Zhao, X., Huang, K.: Globaltrack: A simple and strong baseline for long-term tracking. In: AAAI. vol. 34, pp. 11037–11044 (2020)
18. Huang, Y., Xiao, Z., Firkat, E., Zhang, J., Wu, D., Hamdulla, A.: Spatio-temporal mix deformable feature extractor in visual tracking. Expert Systems with Applications **237**, 121377 (2024)
19. Huang, Y., Li, X., Zhou, Z., Wang, Y., He, Z., Yang, M.H.: Rtracker: Recoverable tracking via pn tree structured memory. In: CVPR. pp. 19038–19047 (2024)
20. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. NIPS **28** (2015)
21. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE TPAMI **34**(7), 1409–1422 (2011)
22. Kang, B., Chen, X., Lai, S., Liu, Y., Liu, Y., Wang, D.: Exploring enhanced contextual information for video-level object tracking. In: AAAI (2025)
23. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: ICCV. pp. 4015–4026 (October 2023)
24. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.K., Danelljan, M., Zajc, L.Č., Lukežič, A., Drbohlav, O., et al.: The eighth visual object tracking VOT2020 challenge results. In: ECCV. pp. 547–601 (2020)
25. Li, X., Zhong, B., Liang, Q., Li, G., Mo, Z., Song, S.: Mambalct: Boosting tracking via long-term context state space model. In: AAAI. vol. 39, pp. 4986–4994 (2025)
26. Liang, S., Bai, Y., Gong, Y., Wei, X.: Autoregressive sequential pretraining for visual tracking. In: CVPR. pp. 7254–7264 (2025)
27. Lin, L., Fan, H., Zhang, Z., Wang, Y., Xu, Y., Ling, H.: Tracking meets lora: Faster training, larger model, stronger performance. In: ECCV. pp. 1–15 (2024)
28. Lin, T.Y., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014)
29. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR. pp. 1–9 (2018)
30. Ma, C., Yang, X., Zhang, C., Yang, M.H.: Long-term correlation tracking. In: CVPR. pp. 5388–5396 (2015)
31. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: ECCV. pp. 445–461 (2016)
32. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: TrackingNet: A large-scale dataset and benchmark for object tracking in the wild. In: ECCV. pp. 300–317 (2018)
33. Pan, X., Yuan, M., Zhang, T., Wang, H.: Deformable attention object tracking network based on cross-correlation. JVIS **98**, 104039 (2024)
34. Peng, L., Gao, J., Liu, X., Li, W., Dong, S., Zhang, Z., Fan, H., Zhang, L.: Vasttrack: Vast category visual object tracking. NeurIPS **37**, 130797–130818 (2024)
35. Shi, L., Zhong, B., Liang, Q., Li, N., Zhang, S., Li, X.: Explicit visual prompts for visual object tracking. In: AAAI. pp. 4838–4846 (2024)
36. Supancic, J.S., Ramanan, D.: Self-paced learning for long-term tracking. In: CVPR. pp. 2379–2386 (2013)
37. Wang, X., Shu, X., Zhang, Z., Jiang, B., Wang, Y., Tian, Y., Wu, F.: Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In: CVPR. pp. 13763–13773 (2021)
38. Wei, X., Bai, Y., Zheng, Y., Shi, D., Gong, Y.: Autoregressive visual tracking. In: CVPR. pp. 9697–9706 (2023)

39. Wu, H., Xu, Z., Zhang, J., Jia, G.: Offset-adjustable deformable convolution and region proposal network for visual tracking. ACCESS **7**, 85158–85168 (2019)
40. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: CVPR. pp. 4794–4803 (2022)
41. Xie, F., Chu, L., Li, J., Lu, Y., Ma, C.: Videotrack: Learning to track objects via video transformer. In: CVPR. pp. 22826–22835 (2023)
42. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: ICCV. pp. 10448–10457 (2021)
43. Yan, B., Zhang, X., Wang, D., Lu, H., Yang, X.: Alpha-refine: Boosting tracking performance by precise bounding box estimation. In: CVPR. pp. 5289–5298 (2021)
44. Yarram, S., Wu, J., Ji, P., Xu, Y., Yuan, J.: Deformable vistr: Spatio temporal deformable attention for video instance segmentation. In: ICASSP. pp. 3303–3307. IEEE (2022)
45. Ye, B., Chang, H., Ma, B., Shan, S., Chen, X.: Joint feature learning and relation modeling for tracking: A one-stream framework. In: ECCV. pp. 341–357 (2022)
46. Zheng, L., Chen, Y., Tang, M., Wang, J., Lu, H.: Siamese deformable cross-correlation network for real-time visual tracking. Neurocomputing **401**, 36–47 (2020)
47. Zheng, Y., Zhong, B., Liang, Q., Mo, Z., Zhang, S., Li, X.: Odtrack: Online dense temporal token learning for visual tracking. In: AAAI. pp. 7588–7596 (2024)
48. Zhou, J., Yao, Y., Yang, R., Xia, Y.: D-transt: Deformable transformer tracking. Electronics **11**(23), 3843 (2022)
49. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR. pp. 9308–9316 (2019)
50. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2020)
51. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: ECCV. pp. 101–117 (2018)