
Dynamic Semantic-Aware Correlation Modeling for UAV Tracking

Xinyu Zhou^{1*} Tongxin Pan^{1*} Lingyi Hong¹ Pinxue Guo² Haijing Guo¹

Zhaoyu Chen² Kaixun Jiang² Wenqiang Zhang^{1,2†}

¹College of Computer Science and Artificial Intelligence, Fudan University

²College of Intelligent Robotics and Advanced Manufacturing, Fudan University
{zhouxinyu20,wqzhang}@fudan.edu.cn, kamipantx@gmail.com

Abstract

UAV tracking can be widely applied in scenarios such as disaster rescue, environmental monitoring, and logistics transportation. However, existing UAV tracking methods predominantly emphasize speed and lack exploration in semantic awareness, which hinders the search region from extracting accurate localization information from the template. The limitation results in suboptimal performance under typical UAV tracking challenges such as camera motion, fast motion, and low resolution, etc. To address this issue, we propose a dynamic semantic aware correlation modeling tracking framework. The core of our framework is a Dynamic Semantic Relevance Generator, which, in combination with the correlation map from the Transformer, explore semantic relevance. The approach enhances the search region’s ability to extract important information from the template, improving accuracy and robustness under the aforementioned challenges. Additionally, to enhance the tracking speed, we design a pruning method for the proposed framework. Therefore, we present multiple model variants that achieve trade-offs between speed and accuracy, enabling flexible deployment according to the available computational resources. Experimental results validate the effectiveness of our method, achieving competitive performance on multiple UAV tracking datasets. The code is available at <https://github.com/zxyxzz/DSATrack>.

1 Introduction

Visual object tracking[55, 9, 29, 28, 59] is a fundamental and important task in computer vision. In recent years, unmanned aerial vehicles (UAV) technology has rapidly developed and demonstrated broad application in various fields such as military, transportation, environmental monitoring, and disaster relief [43]. However, to achieve autonomous task execution in complex environments, object tracking is a crucial component. UAV tracking must address various challenges in dynamic scenarios.

The early UAV tracking methods [36, 31, 35, 27] employed a correlation filter to enhance the discriminative capability, enabling location and tracking. However, these correlation filter-based methods lacked robust representation capabilities. Consequently, UAV tracking methods based on convolutional neural networks (CNNs) [4, 49, 52] emerged, utilizing Siamese network to enhance feature representation, significantly improving tracking accuracy. With the rise of Transformers [55, 11, 6, 51], UAV tracking based on one-stream framework [37, 34, 32, 21] have been studied. Most of the methods focus on improving the efficiency of UAV tracking, while neglecting exploration

*Equal contribution.

†Corresponding author.

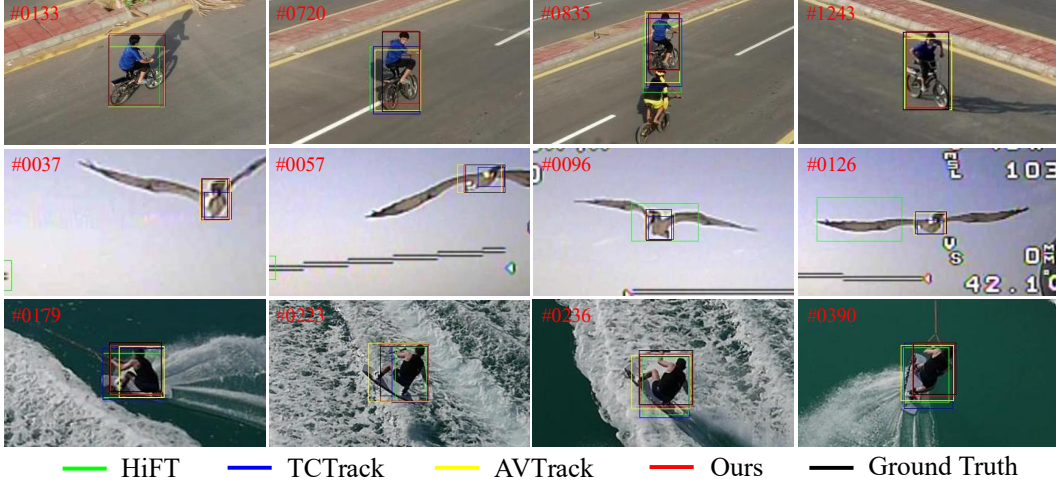


Figure 1: Comparison with SOTA methods under typical UAV challenges. The first to third rows represent camera motion, fast motion, and low resolution, respectively.

of the essence of tracking. Correlation filter-based methods and CNN-based methods rely on local modeling, while Transformer-based methods utilize global modeling. All of the approaches overlook the relevance between semantic features, as shown in Figure 1, leading to suboptimal performance in UAV scenarios, such as camera motion, fast motion, and low resolution, etc.

To address this issue, we propose a dynamic semantic-aware correlation modeling method and develop a UAV tracking framework based on the approach, named DSATrack. Specifically, we use the search region and template&patches as inputs and employ Transformer blocks to achieve semantic matching, semantic awareness, and feature extraction and fusion. Importantly, we design a Dynamic Semantic Relevance Generator within the Transformer to dynamically achieve Semantic Aware Modeling. By leveraging the Semantic Aware Modeling to fuse similar semantic features, our method alleviates semantic ambiguities. The semantic-aware correlation map further facilitates the search region in extracting representative features from the template&patches, thereby enhancing the perception and localization capability of UAV tracking and mitigating tracking drift. Finally, to improve the tracking efficiency, we design a pruning technique tailored to the characteristics of DSATrack. By selectively pruning Transformer blocks, we retain the tracking precision to the greatest extent while achieving high speed tracking. We provides multiple variants for different UAV Tracking application scenarios. For instance, offline tasks such as aerial video prioritize tracking accuracy, while real-time applications like autonomous target following demand fast response times.

Our contributions can be summarized as follows.

- We propose a dynamic semantic-aware UAV tracking framework, DSATrack, which utilizes a Dynamic Semantic Relevance Generator to produce semantic-aware correlation map. These maps are used to fuse features with similar semantic features, effectively reducing tracking drift.
- To further optimize DSATrack, we design a pruning technique that selectively removes Transformer blocks with minimal impact on accuracy. This approach achieves a balance between precision and speed, enabling flexible deployment according to the available computational resources.
- We conduct extensive experiments, demonstrating that DSATrack achieves state-of-the-art performance. Additionally, ablation studies validate the effectiveness of the proposed methods.

2 Related Works

2.1 UAV Tracking

Correlation filter-based methods, such as KCF [25] and DSST [13], were among the earliest applied to UAV tracking due to their high efficiency, achieving strong performance in speed and accuracy.

However, they struggle under camera motion or fast motion, which are common in UAV scenarios. With the rise of deep learning, Siamese network-based trackers—such as TransT [9], AiATrack [19], STARK [53], ToMP [41], and KeepTrack [42]—have significantly improved tracking robustness. TCTrack [4] further adapts Siamese models to UAV tracking. Yet, their heavy computational demands hinder real-time UAV deployment. Recently, one-stream Transformer frameworks [55, 11, 8, 6, 1, 54, 20] have emerged, offering enhanced modeling of long-range dependencies with improved efficiency. Trackers like TATrack [33] and AVTrack [37] utilize this framework to better capture interactions between template and search regions under UAV-specific challenges. Although LoReTrack [14] is not tailored for UAV tracking and lacks real-world UAV evaluation, it demonstrates promising results on a UAV dataset using a lightweight one-stream design. In contrast, we propose a Dynamic Semantic-Aware Correlation Modeling approach within a one-stream Transformer framework to address key UAV challenges such as camera motion, fast motion, and low resolution.

2.2 Semantic Correspondence

In visual perception tasks, semantic correspondence is typically categorized into two types: explicit correspondence and implicit correspondence. Explicit correspondence involves semantic-aware modeling through supervised loss functions, while implicit correspondence does not rely on direct supervision for semantic correspondence. Explicit correspondence methods, such as contrastive learning [7, 24], Triplet Loss [46], and InfoNCE [45], supervise the semantic relationships between images or pixels by pulling closer those belonging to the same category and pushing apart those of different categories. This enables models to learn semantic relationships, improving the accuracy of downstream tasks [22, 58]. On the other hand, implicit correspondence is commonly employed in tasks like classification [40], detection [61], segmentation [10] and Tracking [19, 9] using Transformer-based architectures [48, 15]. These architectures implicitly learn semantic relationships among image features through task-specific losses. Specifically, Transformers compute a correlation map using queries and keys [5, 47], where the correlation map represents the semantic similarity between features [12]. However, in implicit semantic correspondence methods, there is limited exploration into explicitly optimizing semantic correspondence, which often results in suboptimal modeling of the correlation map. This lack of optimization can hinder the model’s ability to capture semantic relationships. Therefore, we propose a dynamic semantic-aware correlation modeling method for UAV tracking to enhance semantic perception capabilities, improving tracking accuracy.

3 Method

In this section, we will first present our proposed tracking framework, DSATrack. Next, we will describe the Dynamic Semantic-aware Transformer. Following that, we will detail the pruning method designed specifically for DSATrack. Finally, we will present description of the prediction head.

3.1 Overall Framework

The overall framework of the proposed DSATrack is shown in Figure 2. The entire tracking framework consists of three components: input representation, feature matching and interaction, and prediction.

Input Representation. The input of DSATrack is the Template & Patches $z \in \mathbb{R}^{n \times 3 \times H_z \times W_z}$ and Search Region $x \in \mathbb{R}^{3 \times H_x \times W_x}$, where H and W represent the height and width of the Template and Search Region. The Template & Patches consist of a fixed template $\in \mathbb{R}^{1 \times H_z \times W_z}$ and several patches $\in \mathbb{R}^{(n-1) \times H_z \times W_z}$ that exhibit representative object characteristics. The Patches are generated and updated over time steps in exactly the same manner as RFGM [60]. Subsequently, the inputs are projected into tokens through a Patch Embedding. **Feature matching and Interaction.** The generated Template Tokens $T_z \in \mathbb{R}^{n \times 3 \times h_z \times w_z}$ and Search Tokens $T_x \in \mathbb{R}^{n \times 3 \times h_x \times w_x}$ are then fed into the Transformer Blocks for feature matching and interaction, where h and w represent the height and width of the features after a $16 \times$ downsampling. The specific computation method of the Transformer block is consistent with OStrack [55]. Importantly, we propose a Dynamic Semantic-aware Transformer for semantic matching and feature fusion. Specifically, we designed a Dynamic Semantic Relevance Generator to construct Semantic Relevance. Using the Semantic Aware Modeling, we fuse the similarities of features with similar semantics to obtain an optimized Semantic-Aware Correlation Map. Based on this correlation map, unimportant template tokens are discarded. The filtered Template Tokens, Search Tokens, and the Semantic-Aware Correlation Map

are then fed into a Hybrid Attention module to facilitate feature interaction. **Prediction.** Finally, the Search Tokens are reassembled into an image and passed to the prediction head, which predicts the bounding box of the object.

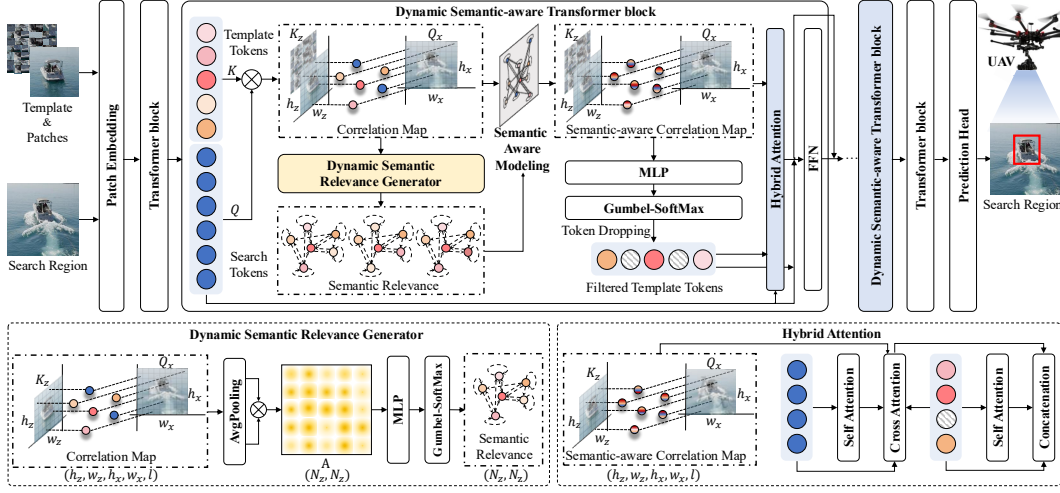


Figure 2: **The framework of DSATrack.** The figure above illustrates the overall pipeline of DSATrack, while the two figures below provide a detailed explanation of the Dynamic Semantic Relevance Generator and the Hybrid Attention, respectively.

3.2 Dynamic Semantic-aware Transformer

The Template Tokens $T_z \in \mathbb{R}^{n \times h_z \times w_z \times d_k}$ and Search Tokens $T_x \in \mathbb{R}^{1 \times h_x \times w_x \times d_k}$ are inputs of Dynamic Semantic-aware Transformer, where d_k is the dimensionality of the features. For simplicity and clarity, we present the details using single template, $T_z \in \mathbb{R}^{1 \times h_z \times w_z \times d_k}$. We use a linear layer to transform the Search Tokens and Template Tokens into ($Q_x \in \mathbb{R}^{h_x \times w_x \times d_k}$) and ($K_z \in \mathbb{R}^{h_z \times w_z \times d_k}$). As shown in Figure 2, we compute the Correlation Map C between the Q_x and K_z as:

$$C_{zx} = \frac{Q_x K_z^\top}{\sqrt{d_k}} \in \mathbb{R}^{h_x \times w_x \times h_z \times w_z \times l}, \quad (1)$$

where C_{zx} captures the correlation scores between query from x and key from z , corresponding to the z -to- x similarity. l represents the number of heads in the Transformer.

In the Correlation Map C_{zx} , the modeling of the similarity relationship between points on Q_x and points on K_z can easily lead to ambiguity. We employ a dynamic perception-based prediction approach to calculate and fuse the similarity between semantically related points on K_z and Q_x .

Dynamic Semantic Relevance Generator. We first reshape $C_{zx} \in \mathbb{R}^{h_x \times w_x \times h_z \times w_z \times l}$ into $C_{zx} \in \mathbb{R}^{N_x \times N_z \times l}$, where $N_x = h_x \times w_x$ and $N_z = h_z \times w_z$. Then, We propose a Semantic Aware Modeling $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\vec{c}_i \mid \vec{c}_i \in C_{zx}, 1 \leq i \leq N_z\}$ represents the set of nodes and the shape of $\vec{c}_i$ is $N_x \times 1 \times l$. In other words, we retrieve nodes along the second dimension of C_{zx} . $\mathcal{E} = \{(u, v) \mid \forall u \in \mathcal{V}, \forall v \in \mathcal{V}\}$ represents the set of Semantic Relevance. Specifically, we apply average pooling to the node features and we concatenate the vector $\vec{c}_i'$ to form $\vec{v}$:

$$\vec{c}_i' = \frac{\sum_{j=0}^{N_x} \vec{c}_j}{N_x}, \vec{c}_j \in \mathbb{R}^{1 \times 1 \times l} \subseteq \vec{c}_i, \quad (2)$$

$$\vec{v} = [\vec{c}_1', \vec{c}_2', \dots, \vec{c}_{N_z}'] \in \mathbb{R}^{N_z \times 1 \times l}, \quad (3)$$

where $[\cdot]$ represents concatenation.

Subsequently, we compute the similarity score A between nodes via matrix multiplication:

$$A = (\vec{v})^\top (\vec{v}), A \in \mathbb{R}^{N_z \times N_z \times l}. \quad (4)$$

After that, a tiny prediction network consisting of a multi-layer perceptron(MLP) and Log-Softmax function is utilized to predict the possibility that each Relevance will be retained and discarded:

$$\pi_{ij} = \log \frac{\exp(\text{MLP}(A_{ij}))}{\sum_{k=1}^{N_z} \exp(\text{MLP}(A_{ik}))} \in \mathbb{R}^2, 1 \leq i, j \leq N_z. \quad (5)$$

where $\pi_{ij,0}$ is the probability of retaining the Relevance (i, j) and $\pi_{ij,1}$ is probability of removing it.

Next, we apply the Gumbel-Softmax to produce the Dynamic Semantic Relevance $\mathcal{E}$, making the π differentiable and retaining the most relevant connections:

$$\mathcal{E} = \text{Gumbel} - \text{Softmax}(\pi) \in \{0, 1\} \in \mathbb{R}^{N_z \times N_z}. \quad (6)$$

Semantic Aware Modeling. Therefore, we utilize the Semantic Aware Modeling to optimize the Correlation Map, as defined by the following formula:

$$C'_{zx} = \Lambda^{-\frac{1}{2}} \hat{\mathcal{E}} \Lambda^{-\frac{1}{2}} C_{zx}^\top W_v \in \mathbb{R}^{N_x \times N_z \times l}, \quad (7)$$

where $\hat{\mathcal{E}} = \mathcal{E} + I$, I is the identity matrix, and $\Lambda_{ii} = \sum_j \mathcal{E}_{ij}$ is the degree matrix.

Hybrid Attention. The Semantic-aware Correlation Map is fed into an MLP and Gumbel-softmax to evaluate the importance of the Template Tokens. Less significant tokens are discarded based on $\mathcal{M} \in \{0, 1\} \in \mathbb{R}^{N_z \times N_z}$ to accelerate computation:

$$\mathcal{M} = \text{Gumbel} - \text{Softmax}(\text{Log} - \text{Softmax}(\text{MLP}(C'_{zx}))). \quad (8)$$

Subsequently, we combine $Q_z, Q_x, K_z, K_x, V_z, V_x$, and C'_{zx} to compute the Hybrid Attention:

$$\begin{aligned} \text{HAttention}(Q_x, K_x, V_x, K_z, V_z) = \\ [\text{SelfAttention}(Q_z, K_z, V_z), \text{SelfAttention}(Q_x, K_x, V_x), \\ \text{CrossAttention}(Q_x, K_z, V_z)]. \end{aligned} \quad (9)$$

The formula for Self-attention is as follows:

$$\text{SelfAttention}(Q, K, V) = \text{SoftMax}(\frac{QK^\top}{\sqrt{d_k}})V. \quad (10)$$

Cross-Attention can be calculated as follows:

$$\text{CrossAttention}(Q_x, K_z, V_z) = \text{SoftMax}(C'_{zx} + C_{zx})V_z. \quad (11)$$

Finally, we incorporated a Feedforward Neural Network (FFN) to enhance the model's fitting ability. The Dynamic Semantic-aware transformer can be summarized as follows:

$$T''_{xz} = T'_{xz} + \text{FFN}(T'_{xz}), \quad (12)$$

$$T'_{xz} = \text{HAttention}(Q_x, K_x, V_x, K_z, V_z) + T_{xz}, \quad (13)$$

where T_{xz} represents the concatenation of T_z and T_x .

3.3 Hierarchical Contribution Ranking Pruning

We introduce the hierarchical contribution metric δ_i , which quantifies the role of each Transformer block in the feature transformation process of DSATrack. The definition of hierarchical contribution is based on the similarity between the hidden state of each layer and its successor. If the input of a layer has a high similarity with the input of the next layer, it indicates that the feature transformation of this layer has a minimal effect on the subsequent layer's input, resulting in a lower contribution. Conversely, if the similarity is low, it suggests that this layer plays a significant role.

The calculation formula for the hierarchical contribution δ_i of the i -th Transformer layer is as follows:

$$\delta_i = 1 - \frac{1}{M} \sum_{m=1}^M \cos(T_{i,m}, T_{i+1,m}) = 1 - \frac{1}{M} \sum_{m=1}^M \frac{T_{i,m} \cdot T_{i+1,m}}{\|T_{i,m}\|_2 \cdot \|T_{i+1,m}\|_2}, \quad (14)$$

where $T_{i,m}$ is the m -th hidden state of i -th Transformer block, and M is the number of hidden states.

The model consists of 12 Transformer blocks, with two types: Dynamic Semantic-Aware Transformer blocks located at layers $\mathcal{D} = \{4, 7, 10\}$, and Standard Transformer blocks in the remaining layers $\mathcal{S} = \{1, 2, 3, 5, 6, 8, 9, 11, 12\}$. For the Dynamic Semantic-Aware Transformer blocks and the Standard Transformer blocks, we first sort the contribution scores δ_i separately for each type of block:

$$\mathcal{C}_{\mathcal{D}} = \text{Sort}(\{\delta_i \mid i \in \mathcal{D}\}), \quad \mathcal{C}_{\mathcal{S}} = \text{Sort}(\{\delta_i \mid i \in \mathcal{S}\}). \quad (15)$$

Based on these sorted values, we select the lowest contributing blocks for pruning. The pruning sets for each type of block are determined by pruning ratios p_d and p_s as follows:

$$\begin{aligned} \mathcal{D}_{\text{prune}} &= \{i \mid i \text{ is the lowest } \lceil p_d \cdot |\mathcal{D}| \rceil \text{ layers in } \mathcal{C}_{\mathcal{D}}\}, \\ \mathcal{S}_{\text{prune}} &= \{i \mid i \text{ is the lowest } \lceil p_s \cdot |\mathcal{S}| \rceil \text{ layers in } \mathcal{C}_{\mathcal{S}}\}. \end{aligned} \quad (16)$$

These pruned blocks are then discarded, and the remaining blocks from both sets are combined to form the new pruned model. The remaining Dynamic Semantic-Aware Transformer blocks and Standard Transformer blocks are:

$$\mathcal{D}_{\text{new}} = \mathcal{D} \setminus \mathcal{D}_{\text{prune}}, \quad \mathcal{S}_{\text{new}} = \mathcal{S} \setminus \mathcal{S}_{\text{prune}}. \quad (17)$$

Finally, the new model’s layer set after pruning is:

$$\mathcal{L}_{\text{new}} = \mathcal{D}_{\text{new}} \cup \mathcal{S}_{\text{new}}. \quad (18)$$

This process allows for a balance between model efficiency and performance, with the pruning ratio p_d and p_s adjustable to optimize computational speed and storage requirements.

3.4 Prediction Head

Following OSTRack [55], we construct our prediction head to estimate the bounding box from the transformer-generated search region features. The head consists of three branches: one predicts the classification score map $S \in [0, 1]^{h_x \times w_x}$, another predicts the offset $O \in [0, 1]^{2 \times h_x \times w_x}$, and the third predicts the normalized box size $E \in [0, 1]^{2 \times h_x \times w_x}$. The location with the highest score, $(x_d, y_d) = \arg \max_{(x, y)} S_{xy}$, is combined with O and E to generate the final bounding box.

$$x = x_d + O(0, x_d, y_d), y = y_d + O(1, x_d, y_d), w = E(0, x_d, y_d), h = E(1, x_d, y_d). \quad (19)$$

Table 1: The hierarchical contribution of DSATrack-ViT-B transformer layers on UAV123 benchmark.

Dataset	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9	Layer 10	Layer 11	Layer 12
UAV123	0.1325	0.0739	0.0615	0.0581	0.0429	0.0457	0.0438	0.0358	0.0428	0.0657	0.1444

Table 2: Setup of removed layers for the pruned variants.

Variant	Removed Layers	Variant	Removed Layers
DSATrack-D8	6, 7, 9, 10	DSATrack-D6	5, 6, 7, 8, 9, 10
DSATrack-D7	6, 7, 8, 9, 10	DSATrack-D4	3, 5, 6, 7, 8, 9, 10, 11

4 Experiments

4.1 Implementation Details

Model. DSATrack employs the ViT-B [15] as the backbone, initialized with MAE [23], referred to as DSATrack-ViT-B. For the pruned variants of DSATrack, we consider four configurations with 8, 7, 6, and 4 transformer layers, named DSATrack-D8, DSATrack-D7, DSATrack-D6, and DSATrack-D4. All pruned models apply the pruning ratio $p_d = \frac{2}{3}$. The pruning ratios p_s are set to $\frac{2}{9}$, $\frac{3}{9}$, $\frac{4}{9}$ and $\frac{6}{9}$ for the four variants. Table 1 presents the hierarchical contribution of DSATrack-ViT-B transformer layers on UAV123, yielding the layer removal setup for pruned model variant, as shown in Table 2. In addition, we provide versions based on ViT-Tiny and DeiT-Tiny, named DSATrack-ViT-T and DSATrack-DeiT-T. Templates are resized to 128×128 , while Search Region are resized to 256×256 .

Training. We train the model on TrackingNet [44], GOT-10k [26], LaSOT [17], and COCO [39]. The training process consists of two stages. In the first stage, the model is trained for 300 epochs with 3 templates. The learning rate for the Dynamic Semantic-aware Transformer and prediction head is set to 4×10^{-4} , the learning rate for the remaining parameters is set to 4×10^{-5} . At epoch 240, the learning rate is decayed by a factor of 10. In the second stage, the model is finetuned for 50 epochs with 6 templates. AdamW optimizer is used with a weight decay of 10^{-4} . The loss function comprises the L1 loss, Focal loss, GIoU loss and ratio loss, consistent with RFGM [60]. All training tasks are conducted on 4 NVIDIA GeForce RTX 3090 GPUs with a batch size of 24.

Inference. For fairness, the speed tests of other models and our model are conducted on the same NVIDIA RTX 3090 GPU without data loading overhead. The size of our Templates & Patches is set to $3 \times N_z$. The update interval of Patches is set to 5 for $t \leq 100$, doubled every 100 frames until $t = 500$, and then remains 160. The interval doubles every 100 frames until it reaches 160, where it remains fixed beyond 160 frames. At the 4th, 7th, and 10th layers, the number of retained template tokens is set to $\lfloor 3 \times N_z \times 0.9 \rfloor$, $\lfloor 3 \times N_z \times 0.8 \rfloor$, and $\lfloor 3 \times N_z \times 0.7 \rfloor$, $\lfloor \cdot \rfloor$ denotes the floor operation.

4.2 State-of-the-Art Comparisons

Figure 5 presents a comparison of DSATrack speed and accuracy against the SOTA methods.

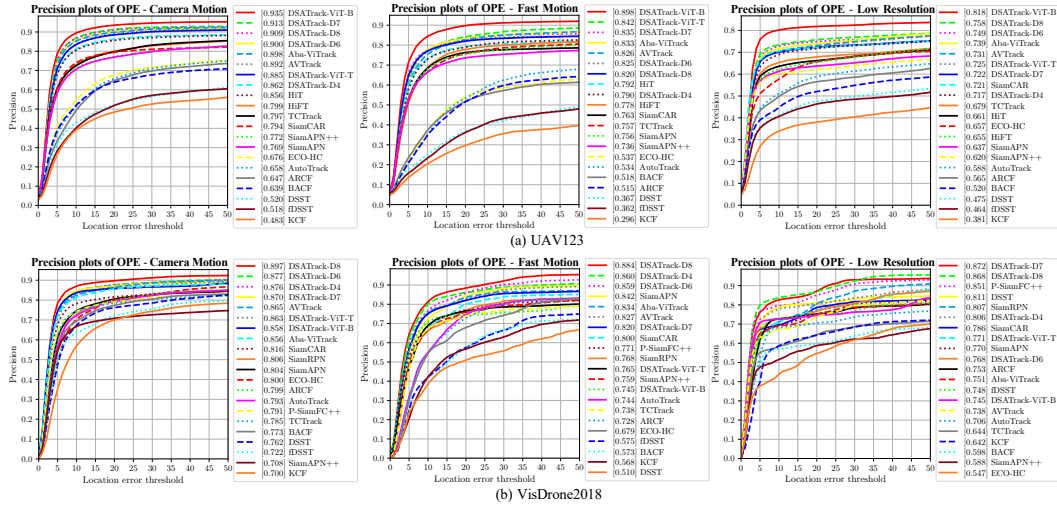


Figure 3: Comparison of state-of-the-art methods under representative UAV challenges.

DTB70. DSATrack outperformed SOTA methods. Specifically, DSATrack-ViT-B attained the highest precision (91.2%) and success (70.6%). Notably, even the lightweight variants, DSATrack-D8 and DSATrack-D7, achieved second (89.4% precision) and third place (88.1% precision), demonstrating the effectiveness of our method across different model complexities. DSATrack-ViT-T acquired 86.6% in Precision and 67.4 % in on success, which is outperform the sota method by 2% to 3%. The results highlight the robustness of dynamic semantic-aware correlation modeling in UAV tracking.

UAVIDT. DSATrack-ViT-B achieved SOTA results with precision and success of 88.1% and 66.3%. DSATrack-D8 also achieved competitive performance, ranking second in precision (85.7%) and achieving the third-best success (63.4%). Importantly, DSATrack-D7 exhibited strong success metrics (62.1%), outperforming prior works such as TATrack and AVTrack. Besides, DSATrack-ViT-T and DSATrack-DeiT-T achieved better performance than SOTA methods, such as AVTrack and LightFC. These results underscore the scalability of DSATrack across varying depths and complexities.

VisDrone2018. DSATrack-D8 and DSATrack-ViT-B achieving the top two rankings for success, 69.1% and 67.0%. D7, ViT-T and DeiT-T achieved competitive results, validating the efficacy of our framework in challenging conditions. Compared to existing SOTA methods, DSATrack consistently presented superior semantic-aware modeling, reflecting its adaptability to UAV scenarios. Moreover, the second row of the Figure 3 demonstrates that the pruned version of DSATrack also exhibits significant advantages in addressing the representative challenges of UAV tracking.

Table 3: Comparisons with SOTA methods on UAV tracking datasets.

Method	Source	DTB70		UAVDT		VisDrone2018		UAV123	
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
SiamAPN[18]	ICRA21	78.4	58.5	71.1	51.7	81.5	58.5	76.5	57.5
SiamAPN++[3]	IROS21	78.9	59.4	76.9	55.6	73.5	53.2	76.8	58.2
HiFT [2]	ICCV21	80.2	59.4	65.2	47.5	71.9	52.6	78.7	58.9
P-SiamFC++[49]	ICME22	80.3	60.4	80.7	56.6	80.1	58.5	74.5	48.9
TCTrack[4]	CVPR22	81.2	62.2	72.5	53.0	79.9	59.4	80.0	60.4
UDAT[56]	CVPR22	80.6	61.8	80.1	59.2	81.6	61.9	76.1	59.0
ABDNet[62]	RAL23	76.8	59.6	75.5	55.3	75.0	57.2	79.3	60.7
DRCI[57]	ICME23	81.4	61.8	84.0	59.0	83.4	60.0	-	-
HiT[30]	ICCV23	75.1	59.2	62.3	47.1	74.8	58.7	82.5	63.3
DDCTrack[16]	ICPR24	79.0	51.1	-	-	81.2	48.7	79.1	50.1
SMAT[21]	WACV24	81.9	63.8	80.8	58.7	82.5	63.4	81.8	64.6
LiteTrack[50]	ICRA24	82.5	63.9	81.6	59.3	79.7	61.4	84.2	65.9
LightFC-ViT[38]	KBS 24	82.8	64.0	83.4	60.6	82.7	62.8	84.2	65.5
AVTrack-ViT[37]	ICML24	81.3	63.3	79.9	57.7	86.4	65.9	84.0	66.2
AVTrack-DeiT[37]	ICML24	84.3	65.0	82.1	58.7	85.9	65.4	84.8	66.8
DSATrack-ViT-B	Ours	91.2	70.6	88.1	66.3	87.1	67.0	90.2	69.4
DSATrack-D8	Ours	<u>89.3</u>	<u>69.5</u>	<u>85.7</u>	<u>63.4</u>	90.8	69.1	<u>87.6</u>	67.0
DSATrack-D7	Ours	88.1	68.4	84.6	62.1	<u>88.6</u>	<u>68.0</u>	87.3	66.6
DSATrack-D6	Ours	86.6	67.3	83.5	60.6	87.0	65.7	86.9	66.0
DSATrack-D4	Ours	84.5	65.9	82.3	58.9	84.8	64.1	84.1	63.0
DSATrack-ViT-T	Ours	86.6	67.4	84.3	61.6	84.8	64.1	85.5	65.5
DSATrack-DeiT-T	Ours	84.5	65.7	85.0	62.0	82.0	60.8	84.4	64.5

UAV123. DSATrack-ViT-B set a new performance of 90.2% (precision) and of 69.4% (success), surpassing existing SOTA trackers. DSATrack-D8/D7 maintaining competitive scores of 87.6% (precision) and 87.3% (success). Furthermore, D4, ViT-T and DeiT-T achieved satisfactory accuracy, making it suitable for real-time UAV applications. As shown in the first row of the Figure 3, DSATrack exhibits better performance in addressing challenges such as camera motion, fast motion, and low resolution. This demonstrates the flexibility of DSATrack in balancing efficiency and accuracy.

4.3 Ablation Study on Correlation Modeling

Table 4: Ablation study on correlation modeling. The best results are highlighted in bold.

Variant	Method	DTB70		UAVDT		VisDrone2018		UAV123	
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
DSATrack-ViT-B	Baseline	87.7	67.9	85.0	63.3	86.5	65.7	87.3	67.2
	Baseline+SSAM	88.0	68.3	85.5	62.6	87.8	66.6	88.8	68.5
	Baseline+DSAM	91.2	70.6	88.1	66.3	87.1	67.0	90.2	69.4

Table 5: Comparisons of Hierarchical Contribution Ranking Pruning and Sequential Pruning.

Backbone	Method	DTB70		UAVDT		VisDrone2018		UAV123		FPS
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	
D7	Sequential Pruning	86.8	67.0	82.9	60.9	86.2	64.0	85.1	64.0	162.1
	Contribution Ranking Pruning	88.1	68.4	84.6	62.1	88.6	68.0	87.3	66.6	162.4
D4	Sequential Pruning	80.6	62.1	77.6	55.0	81.9	60.3	82.8	60.3	210.7
	Contribution Ranking Pruning	84.5	65.9	82.3	58.9	84.8	64.1	84.1	63.0	214.1

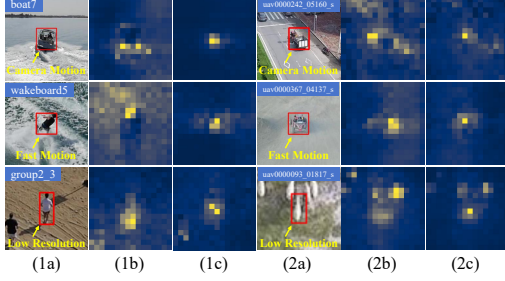


Figure 4: Visualization of correlation map (a) search region. (b) correlation map before the DSAM. (c) correlation map after the DSAM.

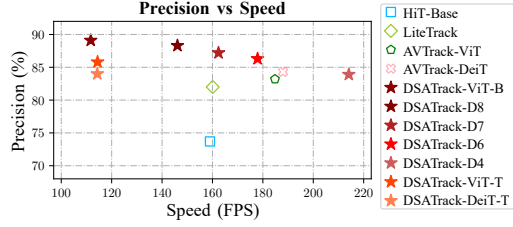


Figure 5: A comparison of speed and accuracy with state-of-the-art methods. The performance is the average across four datasets.

We evaluate our Dynamic Semantic-Aware Modeling (DSAM) against the Baseline and Static Semantic-Aware Modeling (SSAM). Dynamic semantic modeling refers to the dynamic semantic prediction method proposed in this paper, whereas static semantic modeling denotes the approach that applies a 3×3 convolution kernel to the correlation maps from the search region to the template and from the template to the search region, respectively. In this case, the receptive field remains fixed in shape. On DTB70, DSAM with ViT-B achieves 91.% precision and 70.6% success, outperforming both alternatives. Similar trends are observed on UAVDT and UAV123. Unlike SSAM’s fixed-region modeling, DSAM adaptively captures semantic variability, enhancing tracking robustness. As shown in Figure 4, DSAM yields more focused correlation maps, effectively reducing background noise. These results confirm that DSAM significantly improves UAV tracking by modeling dynamic semantic correlations more accurately than static or baseline approaches.

4.4 Ablation Study on Pruning

Sequential Pruning (SP), similar to the proposed Contribution Ranking Pruning (CRP), removes layers 4 and 7 while retaining layer 10. However, SP performs pruning in a fixed sequential order, without explicitly evaluating the relative contribution of each layer to the overall tracking performance. In contrast, CRP introduces a hierarchical contribution ranking mechanism that quantifies each layer’s importance before pruning, ensuring that critical layers are preserved while redundant ones are removed.

As shown in Table 5, CRP consistently outperforms SP across all backbones and datasets. For the D7 variant, CRP achieves higher precision and success rates on all four benchmarks (DTB70, UAVDT, VisDrone2018, and UAV123) with negligible changes in speed. The performance gap becomes even more pronounced in shallower configurations such as D4, where SP suffers from severe accuracy degradation due to over-pruning of essential layers, while CRP maintains stable accuracy with a similar inference speed (214.1 FPS vs. 210.7 FPS).

These results demonstrate that contribution-aware layer selection enables CRP to achieve a better balance between accuracy and computational efficiency. By adaptively ranking layers based on their hierarchical importance rather than following a rigid pruning schedule, CRP achieves a more robust pruning strategy, especially on challenging datasets such as VisDrone2018 and UAV123, where fine-grained spatial details are crucial for tracking performance.

5 Real Word UAV Test

As illustrated in Figure 6, we conduct a series of real-world UAV tracking experiments to further verify the practicality and deployment feasibility of the proposed DSATrack framework. Four representative tracking scenarios are presented, involving diverse targets such as a car, a motorbike, a dog, and a person, covering both rigid and non-rigid motion patterns as well as varying motion speeds and background complexities. These experiments were performed using a Jetson AGX Xavier edge computing platform to emulate realistic onboard conditions with limited computational resources.

To assess the runtime performance, we evaluate the inference speed of different DSATrack variants on the Xavier platform. The lightweight transformer-based variants, DSATrack-ViT-T and DSATrack-DeiT-T, achieve real-time tracking speeds of 27.6 FPS and 27.4 FPS, respectively, demonstrating their

suitability for low-latency applications on embedded hardware. In contrast, larger pruned variants such as DSATrack-D4 and DSATrack-D7 run at approximately 10 FPS, offering a stronger accuracy margin when higher computational capacity is available.

These results highlight the adaptability of DSATrack to diverse deployment scenarios. Depending on the trade-off between accuracy and efficiency, users can flexibly select the appropriate model variant to meet the demands of specific UAV task. The consistent performance across both dynamic and static scenes further confirms the robustness and scalability of the proposed framework in real-world UAV tracking applications.

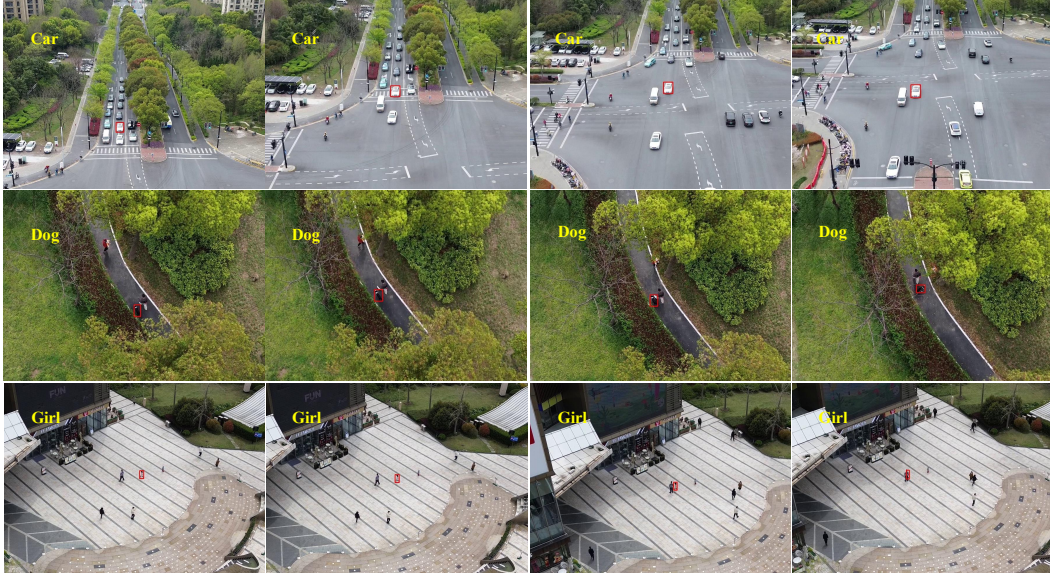


Figure 6: The real-world UAV tracking test includes tracking a car, motorcycle, dog, and person.

6 Conclusion and Limitation

The paper presents DSATrack, a dynamic semantic-aware UAV tracking framework addressing typical UAV Tracking challenges. It incorporates a Dynamic Semantic Relevance Generator to enhance feature fusion and localization, effectively reducing tracking drift. Additionally, a Hierarchical Contribution Ranking Pruning optimizes DSATrack by balancing tracking precision and real-time. Extensive experiments show that DSATrack achieves SOTA performance across various datasets, while ablation studies further validate the effectiveness. DSATrack also has certain limitations. Similar to most trackers, DSATrack is still a local tracker and lacks re-detection capability once the object is lost. In future work, we will explore global UAV tracking algorithms.

Acknowledgement This work was supported by National Natural Science Foundation of China (No.62576109, 62072112).

References

- [1] Yidong Cai, Jie Liu, Jie Tang, and Gangshan Wu. Robust object modeling for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9589–9600, 2023.
- [2] Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. Hift: Hierarchical feature transformer for aerial tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15457–15466, 2021.
- [3] Ziang Cao, Changhong Fu, Junjie Ye, Bowen Li, and Yiming Li. Siamapn++: Siamese attentional aggregation network for real-time uav tracking. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 3086–3092. IEEE, 2021.

- [4] Ziang Cao, Ziyuan Huang, Liang Pan, Shiwei Zhang, Ziwei Liu, and Changhong Fu. Tctrack: Temporal contexts for aerial tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14798–14808, 2022.
- [5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.
- [6] Boyu Chen, Peixia Li, Lei Bai, Lei Qiao, Qihong Shen, Bo Li, Weihao Gan, Wei Wu, and Wanli Ouyang. Backbone is all your need: a simplified architecture for visual object tracking. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 375–392. Springer, 2022.
- [7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [8] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14572–14581, 2023.
- [9] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021.
- [10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [11] Yutao Cui, Cheng Jiang, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13608–13618, 2022.
- [12] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1601–1610, 2021.
- [13] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British machine vision conference, Nottingham, September 1-5, 2014*. Bmva Press, 2014.
- [14] Shaohua Dong, Yunhe Feng, Qing Yang, Yuewei Lin, and Heng Fan. Loretrack: efficient and accurate low-resolution transformer tracking. *arXiv preprint arXiv:2405.17660*, 2024.
- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [16] Guocai Du, Peiyong Zhou, Nurbiya Yadikar, Alimjan Aysa, and Kurban Ubul. Ddctrack: Dynamic token sampling for efficient uav transformer tracking. In *International Conference on Pattern Recognition*, pages 129–144, 2024.
- [17] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Mingzhen Huang, Juehuan Liu, Yong Xu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129(2):439–461, 2021.
- [18] Changhong Fu, Ziang Cao, Yiming Li, Junjie Ye, and Chen Feng. Onboard real-time aerial tracking with efficient siamese anchor proposal network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.
- [19] Shenyuan Gao, Chunlun Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 146–164. Springer, 2022.
- [20] Shenyuan Gao, Chunlun Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18686–18695, 2023.

- [21] Goutam Yelluru Gopal and Maria A Amer. Separable self and mixed attention transformers for efficient object tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6708–6717, 2024.
- [22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.
- [24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [25] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.
- [26] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2019.
- [27] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2891–2900, 2019.
- [28] Kaixun Jiang, Zhaoyu Chen, Jiyuan Fu, Lingyi Hong, Jinglun Li, and Wenqiang Zhang. Videopure: Diffusion-based adversarial purification for video recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.
- [29] Kaixun Jiang, Zhaoyu Chen, Hao Huang, Jiafeng Wang, Dingkan Yang, Bo Li, Yan Wang, and Wenqiang Zhang. Efficient decision-based black-box patch attacks on video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4379–4389, 2023.
- [30] Ben Kang, Xin Chen, Dong Wang, Houwen Peng, and Huchuan Lu. Exploring lightweight hierarchical vision transformers for efficient visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9612–9621, 2023.
- [31] Shui-wang Li, Qian-bo Jiang, Qi-jun Zhao, Li Lu, and Zi-liang Feng. Asymmetric discriminative correlation filters for visual tracking. *Frontiers of Information Technology & Electronic Engineering*, 21(10):1467–1484, 2020.
- [32] Shuiwang Li, Xiangyang Yang, Xucheng Wang, Dan Zeng, Hengzhou Ye, and Qijun Zhao. Learning target-aware vision transformers for real-time uav tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–18, 2024.
- [33] Shuiwang Li, Xiangyang Yang, Xucheng Wang, Dan Zeng, Hengzhou Ye, and Qijun Zhao. Learning target-aware vision transformers for real-time uav tracking. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.
- [34] Shuiwang Li, Yangxiang Yang, Dan Zeng, and Xucheng Wang. Adaptive and background-aware vision transformer for real-time uav tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13989–14000, 2023.
- [35] Shuiwang Li, Qijun Zhao, Ziliang Feng, and Li Lu. Equivalence of correlation filter and convolution filter in visual tracking. In *Image and Graphics: 11th International Conference, ICIG 2021, Haikou, China, August 6–8, 2021, Proceedings, Part III 11*, pages 623–634. Springer, 2021.
- [36] Yiming Li, Changhong Fu, Fangqiang Ding, Ziyuan Huang, and Geng Lu. Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11923–11932, 2020.
- [37] Yongxin Li, Mengyuan Liu, You Wu, Xucheng Wang, Xiangyang Yang, and Shuiwang Li. Learning adaptive and view-invariant vision transformer for real-time uav tracking. In *Forty-first International Conference on Machine Learning*.
- [38] Yunfeng Li, Bo Wang, Xueyi Wu, Zhuoyan Liu, and Ye Li. Lightweight full-convolutional siamese tracker. *Knowledge-Based Systems*, 286:111439, 2024.

- [39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [41] Christoph Mayer, Martin Danelljan, Goutam Bhat, Matthieu Paul, Danda Pani Paudel, Fisher Yu, and Luc Van Gool. Transforming model prediction for tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8731–8740, 2022.
- [42] Christoph Mayer, Martin Danelljan, Danda Pani Paudel, and Luc Van Gool. Learning target candidate association to keep track of what not to track. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13444–13454, 2021.
- [43] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European conference on computer vision*, pages 445–461. Springer, 2016.
- [44] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European conference on computer vision (ECCV)*, pages 300–317, 2018.
- [45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [46] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [47] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [49] Xucheng Wang, Dan Zeng, Qijun Zhao, and Shuiwang Li. Rank-based filter pruning for real-time uav tracking. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06. IEEE, 2022.
- [50] Qingmao Wei, Bi Zeng, Jianqi Liu, Li He, and Guotian Zeng. Litetrack: Layer pruning with asynchronous feature extraction for lightweight and efficient visual tracking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4968–4975. IEEE, 2024.
- [51] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9697–9706, 2023.
- [52] Wanying Wu, Pengzhi Zhong, and Shuiwang Li. Fisher pruning for real-time uav tracking. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022.
- [53] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10448–10457, 2021.
- [54] Dawei Yang, Jianfeng He, Yinchao Ma, Qianjin Yu, and Tianzhu Zhang. Foreground-background distribution modeling transformer for visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10117–10127, 2023.
- [55] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 341–357. Springer, 2022.
- [56] Junjie Ye, Changhong Fu, Guangze Zheng, Danda Pani Paudel, and Guang Chen. Unsupervised domain adaptation for nighttime aerial tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8896–8905, 2022.

- [57] Dan Zeng, Mingliang Zou, Xucheng Wang, and Shuiwang Li. Towards discriminative representations with contrastive instances for real-time uav tracking. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1349–1354. IEEE, 2023.
- [58] Xiangyun Zhao, Raviteja Vemulapalli, Philip Andrew Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label efficient semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10603–10613, 2021.
- [59] Xinyu Zhou, Pinxue Guo, Lingyi Hong, Jinglun Li, Wei Zhang, Weifeng Ge, and Wenqiang Zhang. Reading relevant feature from global representation memory for visual object tracking. *Advances in Neural Information Processing Systems*, 36:10814–10827, 2023.
- [60] Xinyu Zhou, Pinxue Guo, Lingyi Hong, Jinglun Li, Wei Zhang, Weifeng Ge, and Wenqiang Zhang. Reading relevant feature from global representation memory for visual object tracking. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [61] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.
- [62] Haobo Zuo, Changhong Fu, Sihang Li, Kunhan Lu, Yiming Li, and Chen Feng. Adversarial blur-deblur network for robust uav tracking. *IEEE Robotics and Automation Letters*, 8(2):1101–1108, 2023.

A Technical Appendices and Supplementary Material

A.1 Ablation Study on Correlation Modeling on attributes of UAV123.

Table 6: Ablation Study on Correlation Modeling on attributes of UAV123 including Camera Motion, Fast Motion and Low Resolution. Baseline refers to standard Transformer. SSAM represents the static semantic-aware modeling, while DSAM denotes the proposed dynamic semantic-aware modeling. The best results are highlighted in bold.

Variant	Method	Camera Motion		Fast Motion		Low Resolution		Overall	
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
DSATrack-ViT-B	Baseline	90.9	69.9	84.0	64.2	74.2	50.5	87.3	67.2
	Baseline+SSAM	90.6	69.9	84.3	64.5	80.6	55.1	88.8	68.5
	Baseline+DSAM	93.5	72.2	89.8	68.6	81.8	55.9	90.2	69.4

As shown in Table 6, the ablation study demonstrates the effectiveness of Dynamic Semantic-Aware Modeling (DSAM) in enhancing tracking performance across various challenging attributes. Compared to the baseline Transformer, DSAM consistently improves both precision and success rates in all evaluated scenarios. Specifically, for DSATrack-ViT-B, DSAM outperforms the baseline by 2.6% and 2.3% in Camera Motion, 5.8% and 4.4% in Fast Motion, and 7.6% and 5.4% in Low Resolution. This suggests that DSAM effectively captures evolving target representations under complex UAV tracking conditions, where static modeling may struggle to adapt to rapid environmental changes. The performance gap is particularly evident in more challenging scenarios, such as Fast Motion and Low Resolution, where DSAM significantly enhances tracking accuracy and robustness.

To investigate the effect of different template input formats, we compare two variants under the ViT-D8 backbone: using only the template image (A Template) and combining the template with patch embeddings (Template&Patches). As shown in Table 8, incorporating patch-level information consistently improves performance across all UAV tracking benchmarks. Specifically, the Template&Patches design achieves the best overall results, with a precision/success gain of +1.6/+1.6 on UAVDT and +2.3/+1.7 on VisDrone2018 compared to the single-template input. These results demonstrate that introducing fine-grained patch features enhances the representation of template cues, leading to more robust matching and improved generalization across diverse UAV tracking scenarios.

Table 7: Ablation study on the input format of the Template. The best results are highlighted in bold.

Backbone	Method	DTB70		UAVDT		VisDrone2018		UAV123	
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
ViT-D8	A Template	89.5	69.3	85.0	61.8	88.5	67.4	86.4	65.7
	Template&Patches	89.3	69.5	85.2	63.4	90.8	69.1	87.6	67.0

A.2 Ablation study on Token Elimination.

Table 8: Ablation study on token elimination. The best results are highlighted in bold.

Backbone	Method	DTB70		UAVDT		VisDrone2018		UAV123		FPS
		Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	
ViT-Base	w/o token elimination	90.6	69.8	87.3	65.7	86.7	66.3	89.5	67.8	105.6
	w/ token elimination	91.2	70.6	88.1	66.3	87.1	67.0	90.2	69.4	111.7

To evaluate the effectiveness of token elimination, we conduct an ablation study across four UAV tracking benchmarks: DTB70, UAVDT, VisDrone2018, and UAV123. As shown in Table 8, incorporating token elimination consistently improves both tracking precision and success rate across all datasets, demonstrating the robustness and generalizability of the approach. For the ViT-Base backbone, token elimination yields consistent improvements over the baseline that retains all tokens. On DTB70, it achieves a gain of 0.6% in precision and 0.8% in success rate. Similar improvements

Table 9: Comparison with recent transformer-based SOT trackers. AVisT reports AUC/OP₅₀/OP₇₅; TrackingNet and LaSOT report AUC/ P_{norm} / P ; GOT-10k reports AO/SR_{0.5}/SR_{0.75}. “–” denotes not reported.

Method	AVisT			TrackingNet			LaSOT			GOT-10k		
	AUC	OP ₅₀	OP ₇₅	AUC	P_{norm}	P	AUC	P_{norm}	P	AO	SR _{0.5}	SR _{0.75}
SwinTrack	–	–	–	81.1	–	78.4	67.2	70.8	47.6	71.3	81.9	64.5
MixformerV2	–	–	–	83.4	88.1	81.6	70.6	80.8	76.2	–	–	–
AsymTrack	–	–	–	80.0	84.5	77.4	67.7	76.6	61.4	64.7	73.0	67.8
LoRAT-B-224	–	–	–	83.5	87.9	82.1	71.7	80.9	77.3	72.1	81.8	70.7
LoReTrack-256	–	–	–	82.9	81.4	–	70.3	76.2	–	73.5	84.0	70.4
SeqTrack	56.8	66.8	45.6	83.3	88.3	82.2	69.9	79.7	76.3	74.7	84.7	71.8
AQATrack	–	–	–	83.8	88.6	83.1	71.4	81.9	78.6	73.8	83.2	72.1
OSTrack	54.2	63.2	42.2	83.1	87.8	82.0	69.1	78.7	75.2	71.0	80.4	68.2
DSATrack-ViT-B (Ours)	60.2	69.1	50.2	84.1	88.6	82.7	69.4	78.6	74.6	75.0	85.6	73.7

Table 10: Comparison across four UAV tracking benchmarks and efficiency metrics. **Pre.:** precision; **Suc.:** success; FLOPs measured in GigaFLOPs (G).

Method	DTB70		UAVDT		VisDrone2018		UAV123		FLOPs(G)	Params(M)	Xavier(FPS)	RTX3090(FPS)
	Pre.	Suc.	Pre.	Suc.	Pre.	Suc.	Pre.	Suc.				
HiT	75.1	59.2	62.3	47.1	74.8	58.7	82.5	63.3	4.35	42.14	36.6	159.0
AVTrack-ViT	81.3	63.3	79.9	57.7	86.4	65.9	84.0	66.2	1.82	6.20	30.0	184.8
LiteTrack	82.5	63.9	81.6	59.3	79.7	61.4	84.2	65.9	12.81	49.59	15.9	160.1
DSATrack-D7	88.1	68.4	84.6	62.1	88.6	68.0	87.3	66.6	23.64	56.76	14.0	162.4
DSATrack-D4	84.5	65.9	82.3	58.9	84.8	64.1	84.1	63.0	14.12	35.50	17.4	214.1
DSATrack-ViT-T	86.6	67.4	84.3	61.6	84.8	64.1	85.5	65.5	2.99	8.21	27.6	114.3
DSATrack-DeiT-T	84.5	65.7	85.0	62.0	82.0	60.8	84.4	64.5	2.99	8.21	27.4	114.4

are observed on UAVDT (+0.8 precision, +0.6 success), VisDrone2018 (+0.4 precision, +0.7 success), and UAV123 (+0.7 precision, +1.6 success). These gains are particularly significant under challenging scenarios such as camera motion, small object size, and background clutter, which are common in UAV tracking.

Beyond accuracy, token elimination also contributes to runtime efficiency. For instance, with the ViT-Base backbone, the frame rate increases from 105.6 FPS to 111.7 FPS, reflecting reduced computational overhead without sacrificing performance. This efficiency gain makes the model more suitable for real-time UAV deployment, especially on edge computing platforms with limited resources. Overall, the results validate that token elimination effectively filters out redundant or uninformative tokens, allowing the model to focus on semantically relevant features.

A.3 Comparison with General Trackers

Table 9 presents a comprehensive comparison between DSATrack and recent transformer-based single-object trackers across four large-scale benchmarks: AVisT, TrackingNet, LaSOT, and GOT-10k. On AVisT, DSATrack-ViT-B achieves competitive overall performance, obtaining an AUC/OP₅₀/OP₇₅ of 60.2/69.1/50.2, demonstrating superior robustness under adverse visibility and occlusion. On TrackingNet, DSATrack attains the highest AUC of 84.1 and P_{norm} of 88.6, showing strong generalization to large-scale, high-diversity scenes. For LaSOT, DSATrack remains competitive with an AUC of 82.7 and precision of 74.6, comparable to the top-performing AQATrack while requiring no task-specific tuning. On GOT-10k, DSATrack delivers the best accuracy across all metrics (AO/SR_{0.5}/SR_{0.75} = 75.0/85.6/73.7), surpassing previous transformer-based trackers such as OSTrack and LoRAT-B-224.

These results confirm that DSATrack maintains a favorable balance between tracking precision and robustness, matching state-of-the-art transformer trackers across both long-term and general-purpose benchmarks. Its consistent advantage on stricter metrics (OP₇₅, SR_{0.75}) highlights the model’s improved localization accuracy and resilience in challenging conditions.

A.4 Comparison with UAV Trackers

Table 10 compares DSATrack with recent transformer-based UAV trackers in terms of accuracy and efficiency across four representative UAV benchmarks—DTB70, UAVDT, VisDrone2018, and

UAV123—as well as computational complexity metrics. Among all competitors, DSATrack-D7 achieves the best overall accuracy, attaining the highest precision and success scores on every dataset (DTB70: 88.1/68.4, UAVDT: 84.6/62.1, VisDrone2018: 88.6/68.0, UAV123: 87.3/66.6). This indicates that the hierarchical contribution-based pruning and dynamic alignment strategy effectively preserve discriminative representations across diverse UAV scenarios.

Meanwhile, DSATrack-D4 maintains competitive accuracy while significantly improving efficiency, achieving the highest inference speed of 214.1 FPS on RTX 3090 and 17.4 FPS on NVIDIA Xavier, demonstrating excellent scalability for real-time embedded deployment. The lightweight variants DSATrack-ViT-T and DSATrack-DeiT-T further reduce parameters and FLOPs (8.21 M / 2.99 G) while sustaining robust tracking performance, highlighting the flexibility of the proposed framework in balancing accuracy and efficiency across different model scales and hardware settings.

Overall, DSATrack exhibits a favorable accuracy–efficiency trade-off, outperforming existing UAV-specific trackers in both precision and speed, thereby underscoring its practicality for onboard UAV perception applications.

A.5 Broader Impact

The proposed UAV tracking method could benefit public safety and disaster monitoring by enabling real-time, robust tracking from aerial views. However, we also acknowledge the potential for misuse in surveillance applications and stress the importance of deploying such technology within clear legal and ethical boundaries.

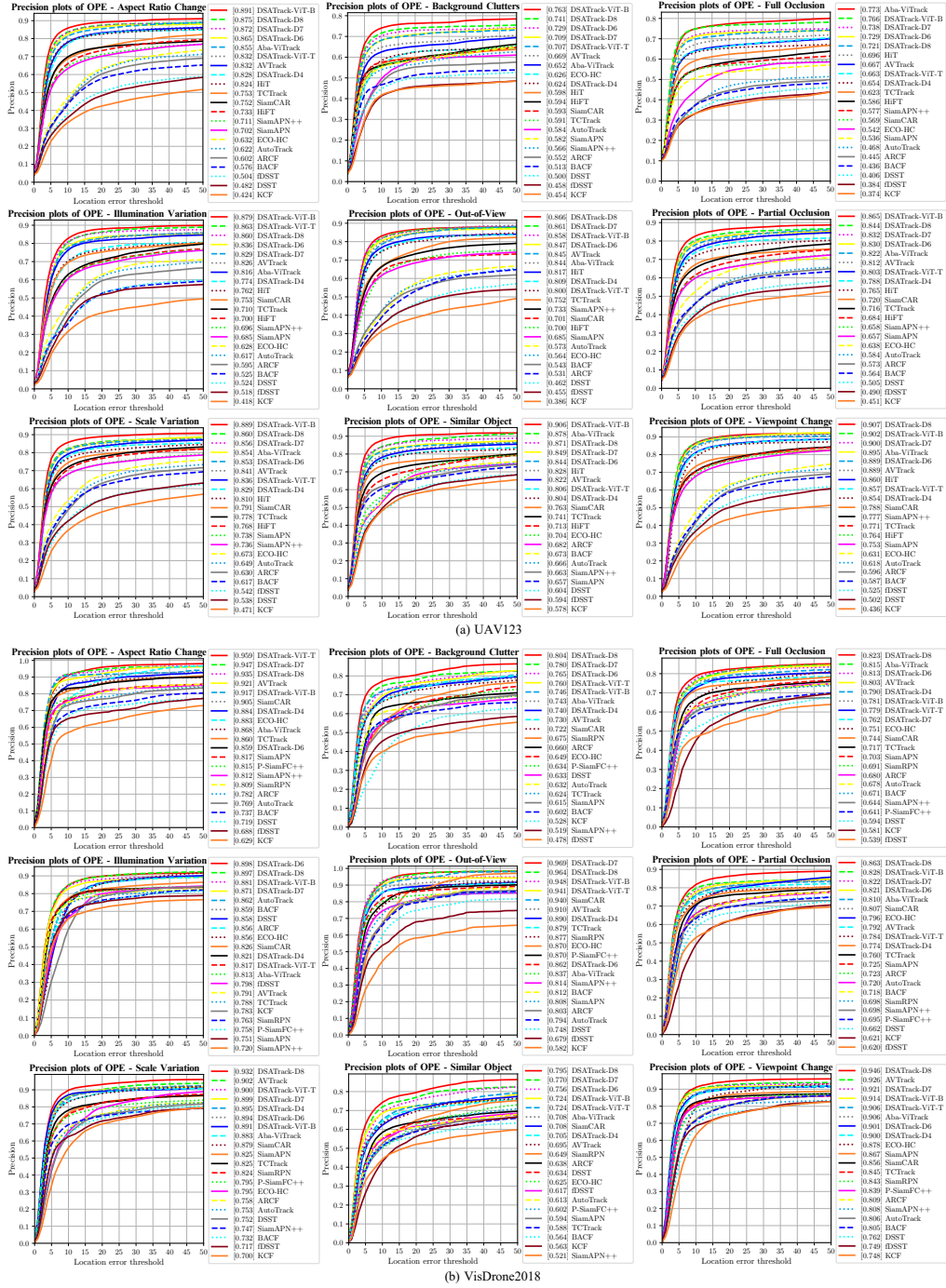


Figure 7: Comparison of state-of-the-art methods under comprehensive UAV challenges.

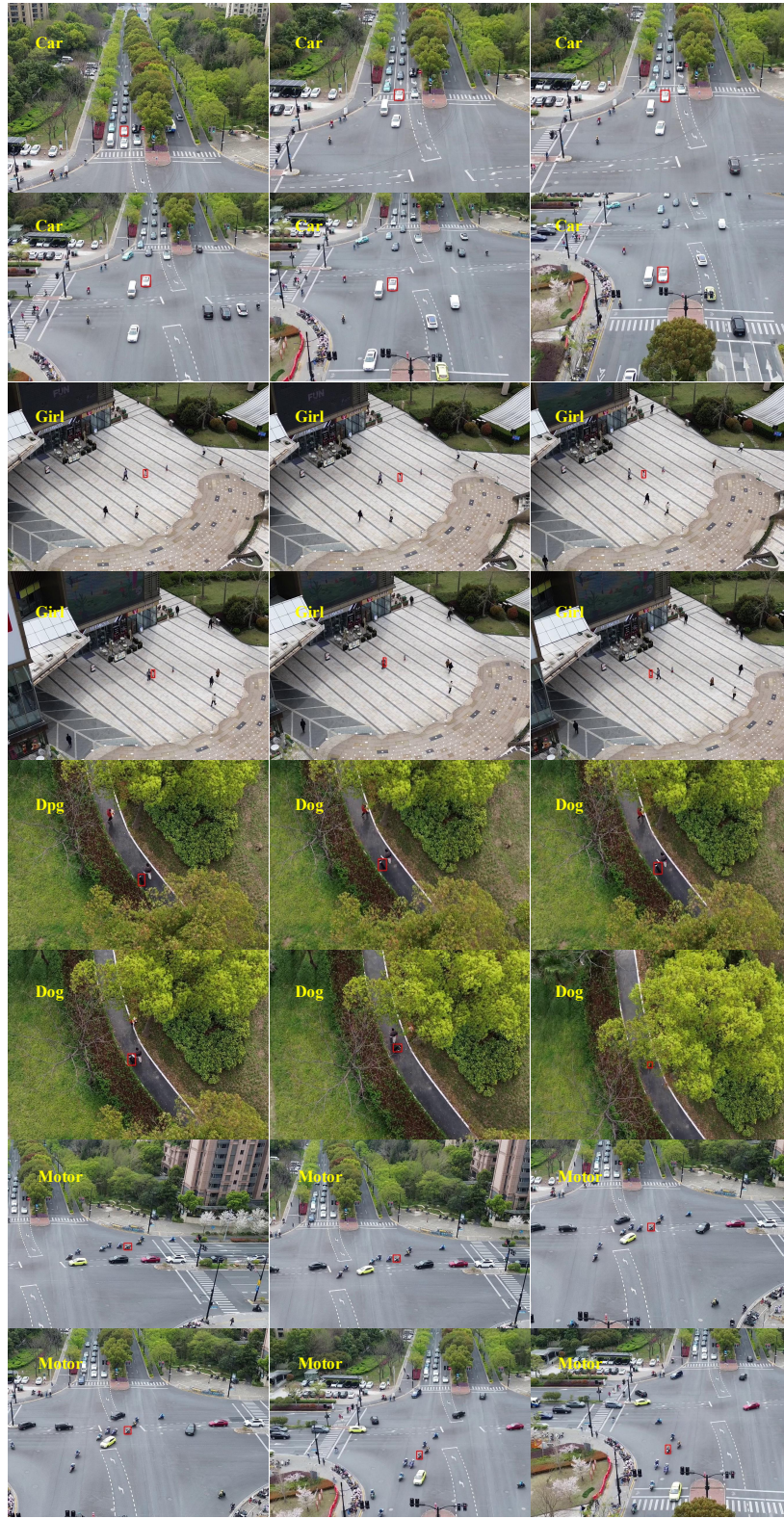


Figure 8: Real-world UAV tracking results on four representative scenarios, including car, motorcycle, dog, and person, demonstrating the adaptability of our method in diverse environments.

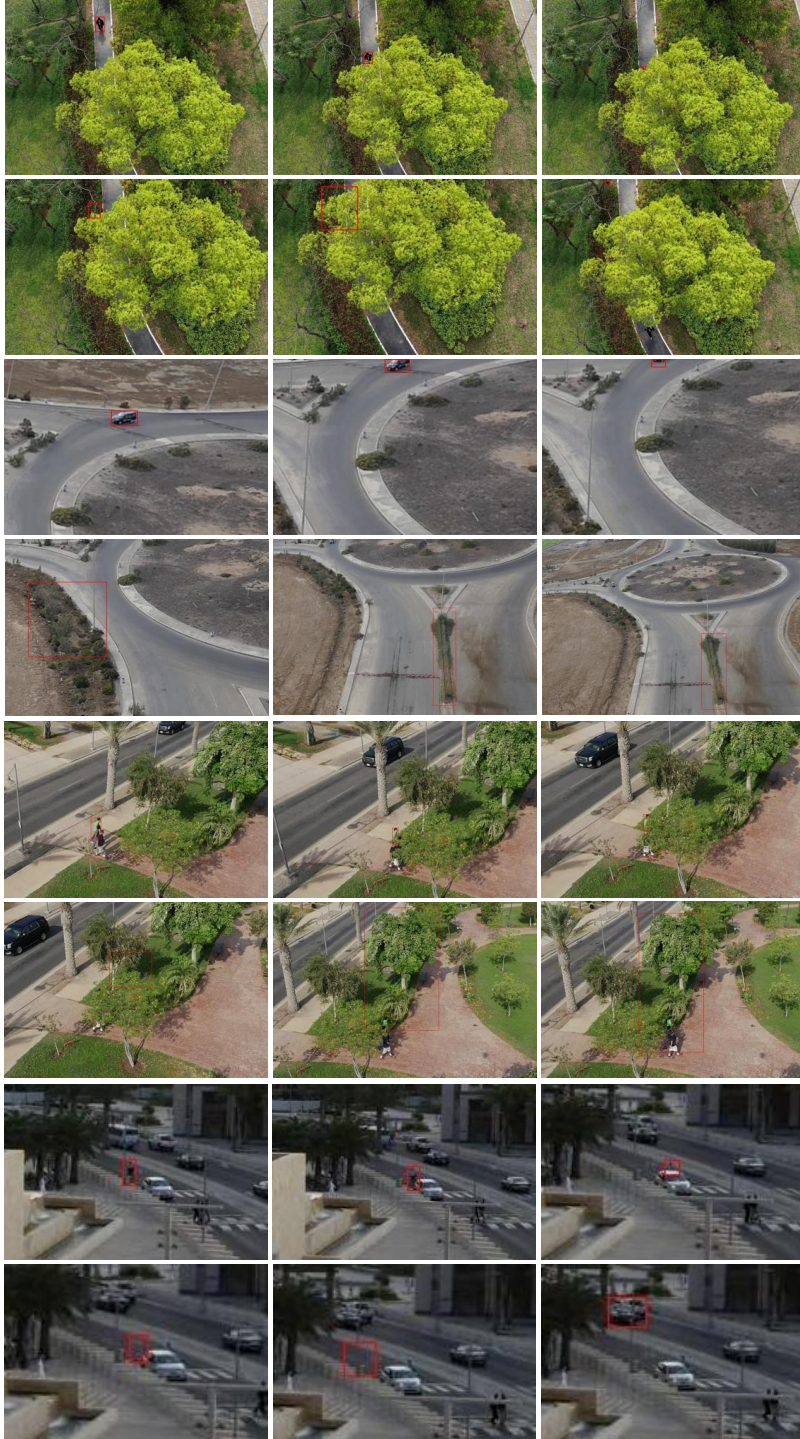


Figure 9: Real-world UAV tracking failure cases on four representative scenarios, including car, motorcycle, dog, and person, demonstrating the adaptability of our method in diverse environments.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction clearly state our contributions.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We have discussed the limitations in section6

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical result,

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [No]

Justification: We will provide opensource code after the paper is accepted.

Guidelines:

- The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We present all details and experimental setting in Section 2 and Section 3

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: We present all experimental setting and detail in Section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: For core experiments, we report precision and success scores across multiple UAV tracking datasets (Table 3), and include performance variance under different conditions (Table 6), reflecting attribute-specific tracking robustness. While error bars are not explicitly drawn, comparative trends are consistently validated.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: We present all details in Section 4 and Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research complies with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We describe the broader impacts in Section 7

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We discuss both the potential positive societal impacts (e.g., public safety, disaster response) and negative ones (e.g., surveillance misuse) of our UAV tracking method in Appendix A.5.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All datasets and model used are public and cited with proper references.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: The paper does not introduce or release any new dataset, all experiments are conducted on existing datasets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not introduce or release any new dataset or model checkpoint; all experiments are conducted on existing datasets.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve human subject research, so no IRB approval is required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No large language models were used in the development of the method or experiments.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.