DASTRACK: EXPLOITING DINOV3 FEATURES, ADAPTIVE MODULATION, AND SPATIAL REFINEMENT FOR VISUAL TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual object tracking has benefited from adopting pre-trained Vision Transformers as backbones. While the recently introduced DINOv3 offers rich intermediate-layer representations, it has not been specifically adapted for tracking, and existing backbone adaptation strategies are insufficient to fully exploit its potential. Moreover, existing feature modulation strategies rely on template-search interaction and coordinate-based trajectory encoding, lacking explicit channel-wise conditioning on target appearance and trajectory modulation that accounts for spatial response distributions. To address these limitations, we propose DASTrack, a framework that exploits DINOv3 features for tracking through dedicated backbone adaptation, adaptive modulation, and spatial refinement. For backbone adaptation, we propose dense LoRA, which jointly encodes template and search images and aggregates multi-layer features, exploiting reliable intermediate representations of the backbone. To explicitly condition features on the target, we introduce template-adaptive modulation and trajectory-guided modulation to enhance search features based on target appearance and historical response distributions, respectively. We further design a spatial refinement head that iteratively refines search features through Gaussian-gated attention to achieve precise localization. We evaluate DASTrack on seven benchmarks, achieving *new* state-of-the-art results of 81.4% AO on GOT-10k and 72.6% AUC on UAV123, with competitive results on other datasets. Code and results will be made publicly available.

1 INTRODUCTION

Visual object tracking is a fundamental task with broad applications, such as autonomous driving, industrial manufacturing, and human-computer interaction (Javed et al., 2022). It aims to continuously localize a target throughout a video sequence, given the bounding box of the target in the initial frame. The quality of backbone features and the effectiveness of post-backbone processing jointly determine tracking accuracy. While recent methods (Cai et al., 2024; Bai et al., 2024; Xie et al., 2025; Kang et al., 2025) have achieved remarkable progress through learning robust target representations and modeling spatio-temporal context, a gap remains between what modern foundation models offer and how existing trackers utilize them.

Recently, self-supervised pre-trained Vision Transformers (ViTs) (Caron et al., 2021; He et al., 2022) have demonstrated strong performance on dense prediction tasks such as segmentation (Hamilton et al., 2022), detection (Li et al., 2022), and depth estimation (Yang et al., 2024), producing high-quality general-purpose visual features. Existing trackers such as LoRAT (Lin et al., 2024), SPMTrack (Cai et al., 2025), and DINO-Tracker (Tumanyan et al., 2024) have employed DINOv2 (Oquab et al., 2024) as their backbone. However, these adaptations focus primarily on parameter efficiency, which is insufficient to fully exploit backbone potential, especially for the reliable intermediate-layer features in DINOv3 (Siméoni et al., 2026).

Beyond backbone adaptation, how to effectively incorporate target-specific refinement and temporal cues into feature processing remains underexplored. Current one-stream frameworks (Wei et al., 2023; Shi et al., 2024) interact template and search features through self-attention at the token level, but do not explicitly condition feature channels on the target appearance. Meanwhile, recent methods have

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

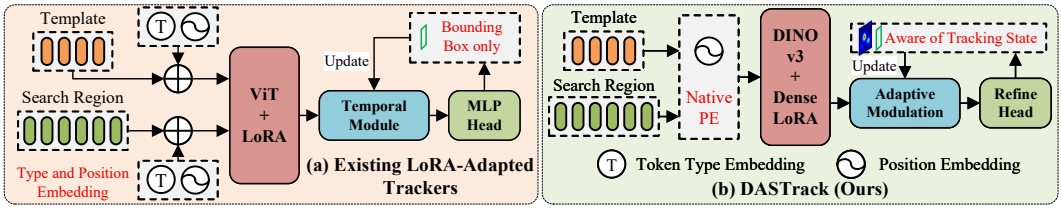

Figure 1: Comparison of (a) existing LoRA-adapted tracking pipelines (Lin et al., 2024; Cai et al., 2025) and (b) our DASTrack. Existing methods introduce additional token type embeddings to distinguish template and search tokens. Instead, DASTrack directly reuses the position embedding of the pre-trained backbone without such embeddings, aggregates reliable intermediate features, and adaptively modulates the search features based on target appearance and historical responses.

incorporated trajectory cues to handle distractors, but rely solely on coordinate-level encoding (Bai et al., 2024; Zhao et al., 2025). Such representations discard the spatial response distribution, which implicitly reveals tracking confidence and the presence of competing candidates. Moreover, most trackers predict the bounding box through a single-pass regression head, leaving no room for iterative refinement under challenging scenarios.

To address these limitations, we propose DASTrack, a framework that exploits DINOv3 features for tracking through dedicated backbone adaptation, adaptive modulation, and spatial refinement (Figure 1). We present Dense LoRA (DLoRA) as a backbone adaptation strategy, which spatially concatenates the template and search region, applies shared positional embeddings and adapters, and aggregates features from multiple intermediate layers rather than relying solely on the final-layer output. To explicitly condition features on the target, we introduce Template-Adaptive Modulation (TAM) and Trajectory-Guided Modulation (TGM). TAM applies a channel-wise affine transformation derived from the template, while TGM spatially gates search features using historical response distributions. Furthermore, we design a Spatial Refinement Head to iteratively refine search features through Gaussian-gated attention in a coarse-to-fine manner. The main contributions are summarized as follows:

- To our best knowledge, we are the first to adapt DINOv3 for visual object tracking. We propose DLoRA as a new backbone adaptation strategy, which directly reuses the pre-trained positional encoding, fine-tunes via lightweight adapters, and exploits reliable intermediate representations.
- To enhance target features and suppress distractors, we introduce template-adaptive modulation for channel-wise affine transformation based on target appearance, and trajectory-guided modulation for spatial feature gating based on historical response distributions.
- To mitigate feature clutter and improve spatial localization, we design a spatial refinement head that iteratively refines search features through Gaussian-gated attention in a coarse-to-fine manner.

2 Related Work

2.1 Visual Tracking with Pre-trained ViTs

Modern trackers have widely adopted pre-trained ViTs (Dosovitskiy et al., 2021; Tian et al., 2024) as backbones. For example, SeqTrack (Chen et al., 2023) and ARTrackV2 (Bai et al., 2024) fully fine-tune MAE-pretrained (He et al., 2022) ViTs, while LoRAT (Lin et al., 2024) and SPMTrack (Cai et al., 2025) adapt DINOv2 (Oquab et al., 2024) through low-rank adaptation (LoRA) (Hu et al., 2022). Beyond backbone adaptation, these methods incorporate additional modules for spatial-temporal modeling (Wei et al., 2023; Zheng et al., 2024; Xie et al., 2025; Zhao et al., 2025). However, DINOv3 (Siméoni et al., 2026) has not yet been systematically adapted for tracking. Specifically, its best-performing features for dense prediction do not necessarily come from the final layer, and its native positional encoding can be shared across spatially concatenated inputs, neither of which is exploited by existing trackers with pre-trained ViTs (Shi et al., 2024; Cai et al., 2024; Kang et al., 2025). We propose Dense LoRA (DLoRA) as a dedicated adaptation strategy that exploits both properties by spatially concatenating inputs and aggregating multi-layer features.

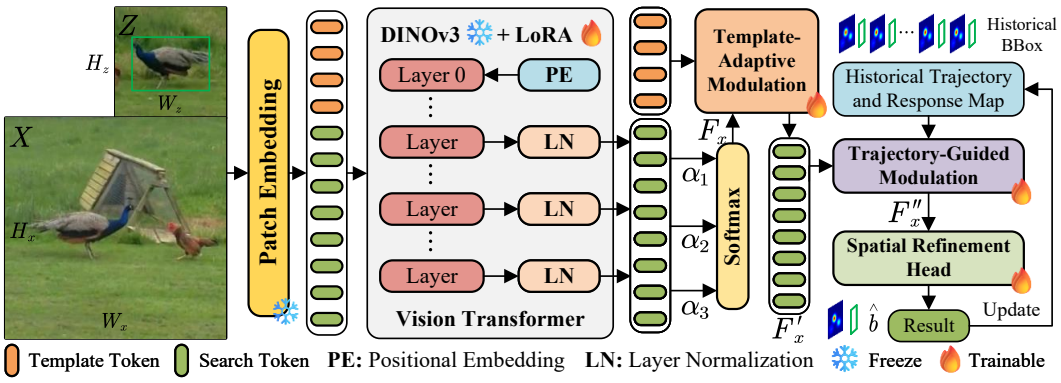

Figure 2: Overall architecture of the proposed tracker. The framework employs a frozen DINOv3 with trainable DLoRA adapters to extract and aggregate multi-layer features. The template-adaptive modulation highlights target-relevant features through channel-wise conditioning, while the trajectory-guided modulation suppresses visually similar distractors using historical tracking states. The spatial refinement head refines search features and the bounding box in a coarse-to-fine manner.

2.2 Feature Modulation and Spatial Refinement

Most existing trackers adopt the one-stream framework (Ye et al., 2022), where template and search tokens interact through symmetric self-attention. Within this framework, several methods apply asymmetric attention (Cui et al., 2022; Bai et al., 2024) or token type embeddings (Lin et al., 2024; Cai et al., 2025; Chen et al., 2025) to differentiate template from search tokens, but these interactions remain at the token level. We propose a different approach to explicitly modulate search features at the channel level based on template appearance. Beyond target appearance, trajectory information offers another source of feature modulation. Several approaches (Xie et al., 2024; Zhao et al., 2025; Wei et al., 2023; Chen et al., 2023) encode historical bounding box coordinates as learnable tokens, but overlook the spatial response distribution that reflects tracking confidence and competing candidates. For the prediction head, current trackers predict the target bounding box through a single-pass regression (Gao et al., 2023; Song et al., 2023; Gao et al., 2022), without explicit spatial priors to guide the prediction process. While refinement modules in ATOM (Danelljan et al., 2019) and Alpha-Refine (Yan et al., 2021) have been proposed, they apply separate networks to refine initial predictions in a post-processing stage. We instead design a Spatial Refinement Head that iteratively refines search features.

3 Method

3.1 Overview

As illustrated in Figure 2, DASTrack takes a template $Z \in \mathbb{R}^{3\times H_z\times W_z}$ and a search region $X \in \mathbb{R}^{3\times H_x\times W_x}$ as input and spatially concatenates them into a single image, where H and W denote the height and width, respectively. The concatenated image is tokenized via patch embedding into $N_z = \frac{H_z}{p} \times \frac{W_z}{p}$ template tokens and $N_x = \frac{H_x}{p} \times \frac{W_x}{p}$ search tokens of dimension C, where p is the patch size. The tokens are then processed by the DINOv3 (Siméoni et al., 2026) backbone with rotary position embedding (RoPE) (Su et al., 2024) and LoRA (Hu et al., 2022) adapters. Next, search tokens are extracted from three selected layers, normalized, and aggregated via learned softmax weights into a unified representation. The aggregated search features are then processed by template-adaptive modulation (TAM), which highlights target-relevant features using template cues. Subsequently, trajectory-guided modulation (TGM) suppresses visually similar distractors using historical tracking states. Finally, the spatial refinement head (SRH) refines the bounding box in a coarse-to-fine manner, yielding the final prediction $\hat{b}$. At each frame, the predicted bounding box and response map are buffered and fed back to TGM for subsequent processing.

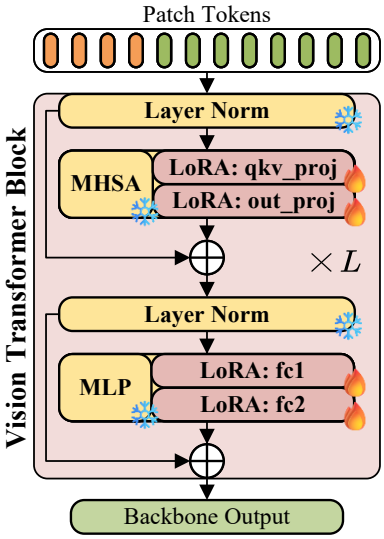

Figure 3: Structure of the DLoRA-adapted ViT block. The low-rank bypasses are attached to all linear layers for task-specific adaptation.

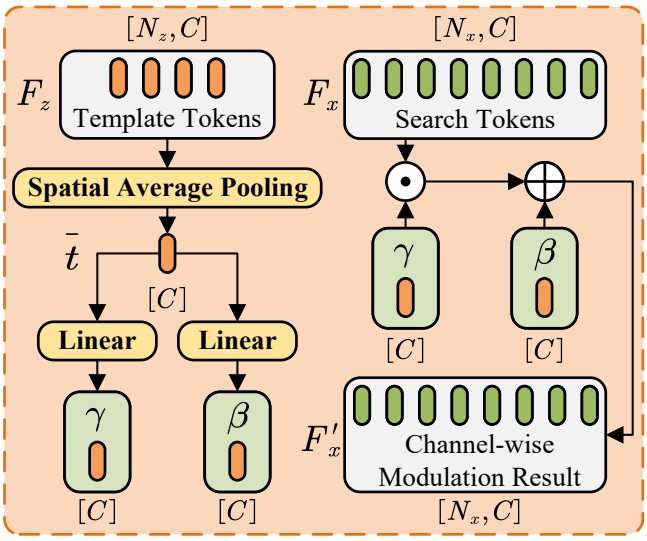

Figure 4: Illustration of template-adaptive modulation. The template tokens are spatially pooled into a global representation, generating per-channel scaling (γ) and shifting (β). The search tokens are then modulated channel-wise by γ and β.

3.2 Dense LoRA Adaptation

Since DINOv3 is pre-trained as a single-image encoder, we spatially concatenate Z and X into a single image, directly reusing its native position embedding. To adapt the DINOv3 backbone for tracking, we adopt LoRA (Hu et al., 2022) to inject trainable low-rank bypasses into frozen linear layers. For each linear layer with frozen weight $W_0 \in \mathbb{R}^{d\times d}$, the output is computed as:

$$y = W_0 h + BAh, \tag{1}$$

where h is the layer input, $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{d\times r}$ are trainable low-rank matrices with rank $r \ll d$, and B is initialized to zero so that the bypass has no effect at the beginning of training. As illustrated in Figure 3, the low-rank bypasses are applied to all linear projections in each Transformer block, including the query-key-value projections and output projections in self-attention, as well as both fully-connected layers in MLP.

Multi-Layer Fusion. While existing trackers (Xie et al., 2025; Zheng et al., 2024; Chen et al., 2023) typically extract features from the final backbone layer, DINOv3 demonstrates that its intermediate layers produce reliable spatial features comparable to the final layer (Siméoni et al., 2026), which motivates multi-layer fusion. We extract search tokens from K selected layers, apply layer normalization, and aggregate them via learnable softmax weights:

$$F_x = \sum_{k=1}^{K} \alpha_k \cdot \mathrm{LN}_k\left(f_k\right), \quad \alpha = \mathrm{Softmax}(a), \tag{2}$$

where $f_k \in \mathbb{R}^{N_x\times C}$ denotes the search tokens from layer k, LN_k is the layer normalization for layer k, and $a \in \mathbb{R}^K$ is a learnable weight vector initialized to zero. We refer to the combination of spatial concatenation, LoRA injection, and multi-layer feature aggregation as Dense LoRA (DLoRA).

3.3 Adaptive Modulation

Template-Adaptive Modulation. In the one-stream tracking framework (Ye et al., 2022), template and search tokens interact through symmetric self-attention, which does not explicitly distinguish their roles. To address this, we propose TAM, which adopts feature-wise linear modulation (Perez et al., 2018) to generate a channel-wise affine transformation from the template (Figure 4). Concretely,

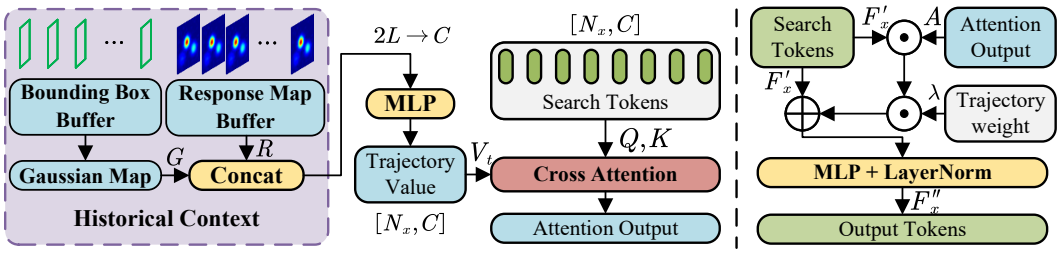

Figure 5: Structure of trajectory-guided modulation. Historical bounding boxes and response maps are encoded into trajectory values, which serve as values in the cross-attention. The result is integrated through a residual connection scaled by a learnable trajectory weight.

given the template tokens $F_z \in \mathbb{R}^{N_z \times C}$ from the final backbone layer and the aggregated search features $F_x \in \mathbb{R}^{N_x \times C}$, TAM pools the template and computes the modulated search features F'_x:

$$\bar{t} = \text{AvgPool}(F_z), \quad \gamma = W_\gamma \bar{t} + b_\gamma, \quad \beta = W_\beta \bar{t} + b_\beta, \quad F'_x = \gamma \odot F_x + \beta, \tag{3}$$

where $\bar{t}, \gamma, \beta \in \mathbb{R}^C$, and $\odot$ denotes channel-wise multiplication. Since γ and β are shared across all spatial positions, TAM amplifies or suppresses feature channels based on the target appearance.

Trajectory-Guided Modulation. While TAM performs channel-wise modulation based on template appearance, TGM spatially modulates features based on historical tracking states (Figure 5). Specifically, TGM maintains two buffers to store the L most recent bounding boxes and response maps. For each normalized historical bounding box $b_l = (c_x^l, c_y^l, w^l, h^l)$, $l = 1, \ldots, L$, where (c_x, c_y) and (w, h) denote the center coordinates and size, anisotropic Gaussian maps are computed as:

$$G^l(x, y) = \exp\left(-\frac{(x - c_x^l)^2}{2(\sigma_x^l)^2} - \frac{(y - c_y^l)^2}{2(\sigma_y^l)^2}\right), \quad (\sigma_x^l, \sigma_y^l) = (w^l, h^l) \cdot \sigma_t, \tag{4}$$

where (x, y) denotes spatial position, and σ_t is a hyperparameter. Then, the Gaussian maps $G \in \mathbb{R}^{N_x \times L}$ and the response maps $R \in \mathbb{R}^{N_x \times L}$ are concatenated along the channel dimension and projected via an MLP, generating the trajectory values $V_t \in \mathbb{R}^{N_x \times C}$. The search features F'_x provide queries and keys in a cross-attention over V_t:

$$V_t = \text{MLP}([G; R]), \quad A = \text{Softmax}\left(\frac{QK^\top}{\sqrt{C}}\right) V_t, \quad F''_x = F'_x + \lambda \cdot (F'_x \odot A), \tag{5}$$

where $Q = W_q F'_x$, $K = W_k F'_x$ are linear projections, and λ is a learnable scalar. The output is further processed by an MLP and layer normalization. By conditioning on historical tracking states, TGM suppresses spatial locations where distractors have appeared in recent frames.

3.4 Spatial Refinement Head

In cluttered scenes, the search features may not be sufficiently discriminative to achieve precise localization. SRH addresses this by refining the search features in a coarse-to-fine manner for M iterations. As summarized in Algorithm 1, the input features F''_x are first assigned to F^0 (Line 1), where the superscript denotes the refinement step. In each step, a center prediction head first produces an intermediate bounding box $\hat{b}_{m-1}$ (Line 3), from which a Gaussian mask is generated (Line 4). In Lines 5–6, the Gaussian mask serves as the value in an attention operation, where the queries and keys are linearly projected from the normalized features F^{m-1} (d denotes the per-head dimension). Subsequently, the attention output is element-wise multiplied with F^{m-1}, normalized, and added back through a

Algorithm 1 Spatial Refinement Head

Input: $F''_x \in \mathbb{R}^{N_x \times C}$, refinement iterations M
Output: Final bounding box $\hat{b}$

1: $F^0 \leftarrow F''_x$
2: **for** $m = 1$ **to** M **do**
3: $\quad \hat{b}_{m-1} \leftarrow \text{CenterHead}_{m-1}(F^{m-1})$
4: $\quad G^m \leftarrow \text{GaussMask}(\hat{b}_{m-1})$
5: $\quad Q, K \leftarrow \text{Linear}(\text{Norm}(F^{m-1}))$
6: $\quad A^m \leftarrow \text{Softmax}(\frac{QK^\top}{\sqrt{d}})\, G^m$
7: $\quad F^m \leftarrow F^{m-1} + \eta_m \cdot \text{Norm}(A^m \odot F^{m-1})$
8: **end for**
9: $\hat{b} \leftarrow \text{CenterHead}_M(F^M)$
10: **return** $\hat{b}$

residual connection (Line 7), where η_m is a learnable scalar. After M steps, a final center prediction head reads out the tracking result $\hat{b}$ from the refined features F^M (Line 9). Unlike the anisotropic Gaussian in TGM (Eq. 4), the Gaussian mask in Line 4 is isotropic:

$$G^m(x,y) = \exp\left(-\frac{(x-c_x^{m-1})^2 + (y-c_y^{m-1})^2}{2(\sigma^m)^2}\right), \quad \sigma^m = \sqrt{w^{m-1}h^{m-1}} \cdot \sigma_h, \tag{6}$$

where σ_h is a hyperparameter.

3.5 Training Loss

We adopt a multi-stage training strategy. We first train the LoRA-adapted backbone jointly with TAM, then incrementally add SRH, TGM, and multi-layer fusion, training each component while freezing all previously trained modules. Since M is the number of refinement iterations in SRH ($M=0$ before SRH is introduced), the training loss of all stages can be uniformly described as:

$$\mathcal{L} = \sum_{m=0}^{M} \left(\mathcal{L}_{\text{cls}}^m + \lambda_{\text{giou}}\,\mathcal{L}_{\text{giou}}^m + \lambda_{L_1}\,\mathcal{L}_{L_1}^m\right). \tag{7}$$

Here, the $\mathcal{L}_{\text{cls}}^m$ is the focal loss (Lin et al., 2017) for classification, while the regression loss comprises the generalized IoU loss (Rezatofighi et al., 2019) $\mathcal{L}_{\text{giou}}^m$ and the L_1 loss $\mathcal{L}_{L_1}^m$. Following the default setting in one-stream trackers (Cai et al., 2024; Guo et al., 2025), the weights $\lambda_{\text{giou}} = 2$ and $\lambda_{L_1} = 5$.

4 Experiments

4.1 Implementation Details

Model Settings. DASTrack employs a frozen DINOv3-L (Siméoni et al., 2026) backbone with LoRA adapters of rank $r=64$. The template and search region are resized to 192×192 and 384×384 pixels, respectively. For multi-layer fusion, we extract search tokens from $K=3$ layers of the 24-layer DINOv3-L backbone, namely layers $[19, 21, 23]$ (0-indexed, layer 23 being the final layer). The SRH (Algorithm 1) applies $M=2$ refinement iterations with $\sigma_h=0.1$ and residual scalars η_m initialized to 0.1. The TGM (Figure 5) buffers the $L=8$ most recent response maps and bounding boxes, with $\sigma_t=0.5$ and residual scalar λ initialized to 0.1. The fusion layers and σ_t are default settings and are adjusted for each benchmark during inference. We adopt OSTrack (Ye et al., 2022) as our baseline.

Training and Inference. We train DASTrack on four datasets, including the training splits of GOT-10k (Huang et al., 2019), LaSOT (Fan et al., 2019), TrackingNet (Muller et al., 2018), and COCO (Lin et al., 2014). For COCO, pseudo video sequences are generated from static images via random translation and scale jittering. We first train the LoRA-adapted backbone jointly with TAM for 300 epochs with a batch size of 16 and 60,000 image pairs per epoch, where the learning rate is 5×10^{-4} for the prediction head and TAM, and 2×10^{-4} for the LoRA adapters. SRH is then added and trained for 100 epochs under the same sampling setting with a batch size of 8. For TGM and multi-layer fusion, we switch to sequential sampling with 16 consecutive frames and 4,000 sequences per epoch, training each for 100 epochs with a batch size of 8 and a learning rate of 2×10^{-4}. All stages use the AdamW optimizer (Loshchilov & Hutter, 2019) and a cosine annealing learning rate scheduler. During inference, DASTrack runs at 18.5 FPS on an NVIDIA RTX 4070Ti GPU.

4.2 Comparisons with the State-of-the-art

GOT-10k (Huang et al., 2019) requires training exclusively on its own training split for fair comparison. As shown in Table 1, DASTrack achieves 81.4% AO, 90.8% $\text{SR}_{0.5}$, and 81.7% $\text{SR}_{0.75}$, setting a new state-of-the-art and outperforming SPMTrack-L and MCITrack-L by 1.4% in AO.

LaSOT (Fan et al., 2019) **and LaSOT$_{\text{ext}}$** (Fan et al., 2021) are large-scale long-term benchmarks. On LaSOT, DASTrack achieves 75.5% AUC, ranking third among all compared methods. On $\text{LaSOT}_{\text{ext}}$, DASTrack achieves 56.0% AUC, 67.9% normalized precision, and 64.5% precision. We further provide attribute-wise analysis on LaSOT (Figure 6). Compared to LoRAT-L, which also adapts DINOv2 via LoRA, DASTrack shows improvements on illumination variation (+1.6%) and out-of-view (+1.0%), which we attribute to the explicit conditioning on target appearance (TAM) and

Table 1: State-of-the-art comparison on the GOT-10k, LaSOT, LaSOT$_{ext}$, TrackingNet datasets. The top two results are highlighted in red and blue, respectively. † denotes using additional VastTrack (Peng et al., 2024) training data.

Method	GOT-10k (Huang et al., 2019)			LaSOT (Fan et al., 2019)			LaSOT$_{ext}$ (Fan et al., 2021)			TrackingNet (Muller et al., 2018)		
	AO(%)	$SR_{0.5}$(%)	$SR_{0.75}$(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)	AUC(%)	P_{Norm}(%)	P(%)
Ours	**81.4**	**90.8**	**81.7**	**75.5**	**85.8**	**84.0**	**56.0**	**67.9**	**64.5**	**86.4**	**90.6**	**87.2**
Δ (vs OSTrack384)	(+7.7)	(+7.6)	(+10.9)	(+4.4)	(+4.7)	(+6.4)	(+5.5)	(+6.6)	(+6.9)	(+2.5)	(+2.1)	(+4.0)
SPMTrack-L (Cai et al., 2025)	**80.0**	**89.4**	**79.9**	**76.8**	**85.9**	**84.0**	-	-	-	**86.9**	**91.0**	**87.2**
MCITrack-L† (Kang et al., 2025)	**80.0**	**88.5**	**80.2**	**76.6**	**86.1**	**85.0**	**55.7**	**66.5**	**62.9**	**87.9**	**92.1**	**89.2**
MambaLCT-384 (Li et al., 2025)	76.2	86.7	74.3	73.6	84.1	81.6	53.3	64.8	61.4	85.2	89.8	85.2
TemTrack (Xie et al., 2025)	76.1	84.9	74.4	73.1	83.0	80.7	53.4	64.8	61.0	85.0	89.3	84.8
ARTrackV2-L384 (Bai et al., 2024)	**79.5**	87.8	79.6	73.6	82.8	81.1	53.4	63.7	60.2	86.1	90.4	86.2
ODTrack-L (Zheng et al., 2024)	78.2	87.2	77.3	74.0	84.2	**82.3**	53.9	65.4	61.7	86.1	**91.0**	**86.7**
RTracker-L (Huang et al., 2024)	77.9	87.0	76.9	74.7	84.5	-	54.9	65.5	62.7	-	-	-
LoRAT-L (Lin et al., 2024)	77.5	86.2	78.1	75.1	84.1	82.0	**56.6**	**69.0**	**65.1**	85.6	89.7	85.4
HIPTrack (Cai et al., 2024)	77.4	88.0	74.5	72.7	82.9	79.5	53.0	64.3	60.6	84.5	89.1	83.8
EVPTrack-384 (Shi et al., 2024)	76.6	86.7	73.9	72.7	82.9	80.3	53.7	65.5	61.9	84.4	89.1	-
AQATrack-384 (Xie et al., 2024)	76.0	85.2	74.9	72.7	82.9	80.2	52.7	64.2	60.8	84.8	89.3	84.3
ARTrackL384 (Wei et al., 2023)	78.5	87.4	77.8	73.1	82.2	80.3	52.8	62.9	59.7	85.6	89.6	86.0
DropTrack (Wu et al., 2023)	75.9	86.8	72.0	71.8	81.8	78.1	-	-	-	84.1	88.9	-
ARTrack384 (Wei et al., 2023)	75.5	84.3	74.3	72.6	81.7	79.1	51.9	62.0	58.5	85.1	89.1	84.8
SeqTrack-L384 (Chen et al., 2023)	74.8	81.9	72.2	72.5	81.5	79.3	50.7	61.6	57.5	85.5	89.8	85.8
ROMTrack-384 (Cai et al., 2023)	74.2	84.3	72.4	71.4	81.4	78.2	51.3	62.4	58.6	84.1	89.0	83.7
GRM (Gao et al., 2023)	73.4	82.9	70.4	69.9	79.3	75.8	-	-	-	84.0	88.7	83.3
CTTrack-L (Song et al., 2023)	72.8	81.3	71.5	69.8	79.7	76.2	-	-	-	84.9	89.1	83.5
TATrack-L (He et al., 2023)	-	-	-	71.1	79.1	76.1	-	-	-	85.0	89.3	84.5
MixViT-L (Cui et al., 2022)	75.7	85.3	75.1	72.4	82.2	80.1	-	-	-	85.4	90.2	85.7
OSTrack384 (Ye et al., 2022)	73.7	83.2	70.8	71.1	81.1	77.6	50.5	61.3	57.6	83.9	88.5	83.2
AiATrack (Gao et al., 2022)	69.6	80.0	63.2	69.0	79.4	73.8	47.7	55.6	55.4	82.7	87.8	80.4

Table 2: State-of-the-art comparison on the TNL2K, UAV123, and NFS30 datasets. The top two results are highlighted in red and blue, respectively.

Method	TNL2K AUC(%)	UAV123 AUC(%)	NFS30 AUC(%)
DASTrack (Ours)	**63.3**	**72.6**	**69.7**
Δ (vs OSTrack384)	(+7.4)	(+1.9)	(+3.2)
MCITrack-L† (Kang et al., 2025)	**65.3**	71.5	**70.6**
LoRAT-L (Lin et al., 2024)	**62.3**	**72.5**	66.7
SPMTrack-B (Cai et al., 2025)	62.0	**71.7**	67.4
ARTrackV2-L384 (Bai et al., 2024)	61.6	**71.7**	**68.4**
ARTrackL384 (Wei et al., 2023)	60.3	71.2	67.9
AQATrack-384 (Xie et al., 2024)	59.3	71.2	-
EVPTrack-384 (Shi et al., 2024)	59.1	70.9	-
TemTrack (Xie et al., 2025)	58.8	70.8	-
HIPTrack (Cai et al., 2024)	-	70.5	68.1
MambaLCT-384 (Li et al., 2025)	58.5	70.1	-
SeqTrack-L384 (Chen et al., 2023)	57.8	68.5	66.2
OSTrack384 (Ye et al., 2022)	55.9	70.7	66.5

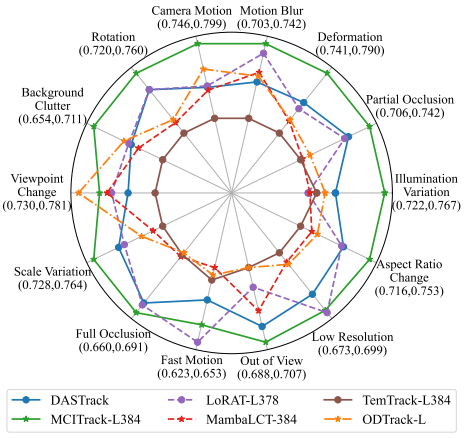

Figure 6: Attribute-wise results on LaSOT.

historical tracking states (TGM). However, DASTrack lags behind LoRAT-L on fast motion (−1.7%) and motion blur (−1.5%), suggesting room for improvement in handling the fast motion scenario.

TrackingNet (Muller et al., 2018) **and TNL2K** (Wang et al., 2021) are challenging benchmarks across diverse scenarios. In Tables 1 and 2, DASTrack achieves 86.4% AUC on TrackingNet and 63.3% on TNL2K, outperforming the DINOv2-based tracker LoRAT-L by 0.8% and 1.0%, respectively.

UAV123 (Mueller et al., 2016) **and NFS30** (Kiani Galoogahi et al., 2017) are classical benchmarks for unmanned aerial vehicle (UAV) tracking and fast-motion scenarios. As shown in Table 2, DASTrack achieves 72.6% AUC on UAV123 and 69.7% on NFS30, setting a new state-of-the-art on UAV123 and surpassing LoRAT-L by 3.0% on NFS30.

4.3 Ablation Studies

Effect of the Proposed Components. We progressively add each module to the LoRA-adapted DINOv3 baseline (Table 3). TAM and TGM each yield +0.7% AO on GOT-10k, the largest per-module gains, confirming that conditioning on target appearance and tracking history effectively enhances feature discrimination. SRH contributes +0.6% on GOT-10k and +0.1% on TNL2K through iterative spatial refinement. The multi-layer fusion (MLF) improves TNL2K by 0.3% but slightly

Table 3: Effect of the proposed components on GOT-10k and TNL2K. MLF: multi-layer fusion.

#	TAM	SRH	TGM	MLF	GOT-10k AO(%)	TNL2K AUC(%)
1					79.4	61.8
2	✓				80.1 (+0.7)	62.4 (+0.6)
3	✓	✓			80.7 (+0.6)	62.5 (+0.1)
4	✓	✓	✓		**81.4** (+0.7)	63.0 (+0.5)
5	✓	✓	✓	✓	81.1 (-0.3)	**63.3** (+0.3)

Table 4: Comparison of self-supervised pretrained backbones on GOT-10k.

#	Pretrained Model	Params (M)	MACs (G)	GOT-10k AO(%)
1	MAE ViT-B/16	86.4	50.4	73.7 (OSTrack)
2	DropMAE ViT-B/16	86.1	55.5	75.9 (DropTrack)
3	MAE ViT-L/16	303.7	191.2	74.8 (SeqTrack)
4	Fast-iTPN-L/16	311.5	207.2	80.0 (MCITrack)
5	DINOv2 ViT-L/14	303.7	247.3	77.5 (LoRAT)
6	**DINOv3-L/16**	303.1	192.7	**80.1**

Table 5: Ablation results of TGM on GOT-10k. Res. denotes the residual connection in Eq. 5, versus $F''_x = F'_x \odot A$. Here, λ is the modulation strength, which can be a single learnable scalar, a learned per-token gate, or a per-frame weight supervised by the ground-truth IoU.

#	Res.	Trajectory Input	λ	AO(%)
1	✗	Gaussian Map	—	80.1
2	✓	Gaussian Map + Cls Score	scalar	80.9
3	✓	Gaussian Map + Response Map	scalar	**81.4**
4	✓	Gaussian Map + Response Map	learned gate	80.1
5	✓	Gaussian Map + Response Map	IoU-supervised	80.5

Table 6: Ablation results of SRH on GOT-10k. Two attention formulations are compared: G as the attention value with element-wise feature modulation, or G weights the attention logits with F as the value.

#	Attention	Gaussian	Res.	AO(%)
1	$\mathrm{Attn}(Q,K,G)\odot F$	isotropic	✗	77.8
2	$\mathrm{Attn}(Q,K,G)\odot F$	isotropic	✓	**80.7**
3	$\mathrm{Attn}(Q,K,G)\odot F$	anisotropic	✓	80.4
4	$\mathrm{Attn}(Q,K\odot G,F)$	isotropic	✓	80.4
5	$\mathrm{Attn}(Q,K\odot G,F)$	anisotropic	✓	80.6

decreases GOT-10k by 0.3%. We attribute this to the GOT-10k protocol, which restricts training to a single dataset where the well-adapted final-layer features are already sufficiently discriminative, while MLF benefits from the larger and more diverse full training set used for TNL2K.

Effect of Pretrained Backbone. Table 4 compares different self-supervised pretrained backbones used in recent trackers on GOT-10k. With only the LoRA-adapted DINOv3 backbone and TAM, our tracker already achieves 80.1% AO, outperforming the DINOv2-based LoRAT by 2.6% with a lower backbone computational cost and comparable backbone parameter count.

Effect of Trajectory-Guided Modulation. Table 5 ablates the residual connection, trajectory input, and modulation strength λ in TGM. Replacing spatial response maps with scalar classification scores decreases AO by 0.5% (Lines 2 and 3), confirming that spatial response distributions carry richer information than scalar scores. For λ, both adaptive gating variants (Lines 4 and 5) underperform the learnable scalar in Line 3. This is likely because the attention mechanism already provides spatially adaptive modulation, making an additional adaptive λ redundant.

Effect of Spatial Refinement Head. Table 6 ablates the attention formulation, Gaussian shape, and residual connection in SRH. Removing the residual connection (Lines 1 and 2) causes a 2.9% AO drop, making it the most critical design choice. The four combinations of attention formulation and Gaussian shape (Lines 2–5) vary within 0.3% AO, suggesting that SRH is robust to both choices.

Effect of LoRA Rank. Table 7(a) varies the LoRA rank r on LaSOT. Small ranks $r=16$ and $r=32$ lack sufficient capacity for adaptation, while $r=128$ brings no further improvement. The default setting $r=64$ provides a good balance between adaptation capacity and parameter efficiency.

Effect of Different Fusion Layers. Table 7(b) compares different layer combinations for multi-layer fusion. All four configurations perform within 81.0–81.1% AO, indicating that the fusion is robust to layer selection on GOT-10k. We adopt $[19,\ 21,\ 23]$ as the default, which restricts the fusion to the later part of the network, in line with the per-layer analysis of DINOv3 (Appendix A.1).

Effect of Refinement Iterations. Table 7(c) varies the number of refinement iterations M in SRH. Performance improves steadily from $M=0$ to $M=2$, confirming the benefit of iterative spatial refinement. $M=3$ drops slightly to 80.4%, indicating diminishing returns beyond two iterations.

Sensitivity of Gaussian Spread. Table 8(a–b) varies the Gaussian spread σ_h in SRH and σ_t in TGM. The model is robust to σ_h within $[0.05, 0.20]$, where GOT-10k AO varies within 0.4%, while UAV123

Table 7: Architecture hyperparameters analysis on LaSOT and GOT-10k.

(a) LoRA Rank		(b) Fusion Layers N_f		(c) Iterations M	
r	LaSOT AUC(%)	N_f	GOT-10k AO(%)	M	GOT-10k AO(%)
16	74.6	[7, 15, 23]	**81.1**	0	80.1
32	74.7	[15, 19, 23]	81.0	1	80.5
64	**75.1**	**[19, 21, 23]**	81.0	**2**	**80.7**
128	74.5	[23, 23, 23]	81.0	3	80.4

Table 8: Sensitivity analysis on GOT-10k and UAV123.

(a) SRH σ_h			(b) TGM σ_t			(c) Buffer L		
σ_h	GOT-10k AO(%)	UAV123 AUC(%)	σ_t	GOT-10k AO(%)	UAV123 AUC(%)	L	GOT-10k AO(%)	UAV123 AUC(%)
0.05	80.3	71.62	0.1	79.5	**72.61**	2	80.8	72.13
0.10	**80.7**	**71.84**	0.3	80.3	71.97	4	79.9	72.19
0.15	80.4	71.81	**0.5**	**80.9**	72.22	**8**	**80.9**	**72.22**
0.20	80.4	71.76	0.7	79.1	71.52	12	80.3	72.19
0.30	79.5	71.81	0.9	78.9	70.90	16	79.3	72.08

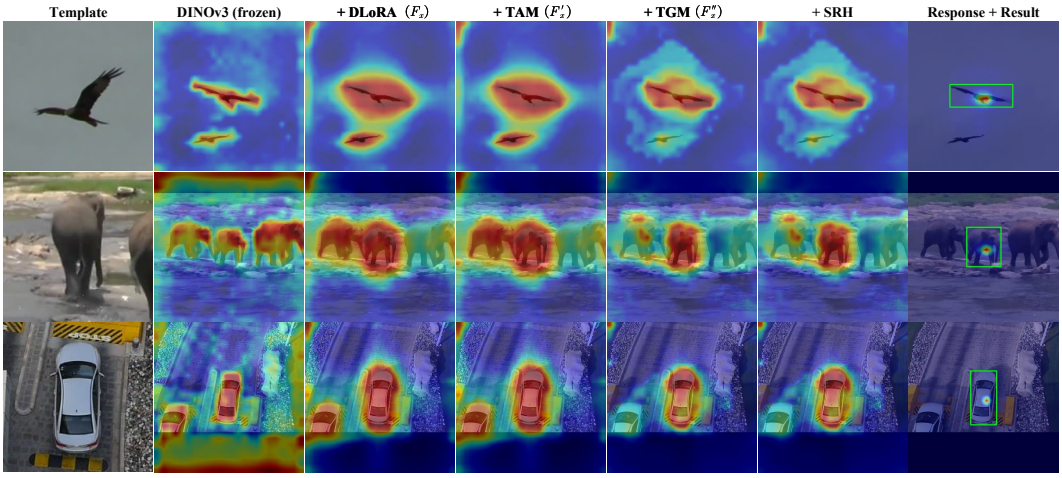

Figure 7: Visualization of search-region features at each processing stage on three scenes with visually similar distractors. Each column shows the feature map after adding the labeled module. Since each map is independently normalized, the channel-wise scaling effect of TAM is not visually apparent. The last column overlays the final response map and predicted bounding box.

AUC varies by less than 0.3% across the full range. For σ_t, GOT-10k AO peaks at $\sigma_t = 0.5$ with a sharper drop at both extremes, while UAV123 AUC favors smaller values.

Effect of Buffer Length. Table 8(c) varies the TGM buffer length L at inference time by linearly interpolating the historical bounding boxes and response maps to the target length. On GOT-10k, $L = 8$ achieves the best AO of 80.9%, with performance dropping at both shorter and longer buffers. On UAV123, AUC is largely insensitive to L, varying by only 0.14% across the full range.

Feature Visualization. As shown in Figure 7, the frozen DINOv3 features respond broadly to semantically similar regions without distinguishing the target from distractors, while the activations become more structured and concentrated on object regions after adding DLoRA. TAM reweights the channel-wise feature importance based on the template, but its effect is not visually apparent, since the spatial activation pattern is largely preserved. After TGM, distractor activations are visibly attenuated while the target region remains highlighted. SRH further sharpens the activation around the target. The final response map peaks sharply on the target, yielding precise bounding-box predictions.

5 Conclusion

We presented DASTrack, a tracker that exploits DINOv3 features through Dense LoRA adaptation, adaptive modulation, and iterative spatial refinement. Starting from a LoRA-adapted DINOv3 baseline, TAM, TGM, and SRH bring gains of 2.0% AO on GOT-10k and 1.5% AUC on TNL2K, indicating that LoRA fine-tuning alone does not fully utilize DINOv3 features for tracking. Compared with the DINOv2-based LoRAT-L, DASTrack achieves an average gain of 1.2% across *seven* benchmarks. Nevertheless, DASTrack remains less robust under fast motion, where the tracking targets with large displacements may move away from both the search region and the historical trajectory location. Future work may explore motion-aware search region cropping and trajectory priors that adapt to large target displacements, as well as more efficient backbone adaptation.

AI USE STATEMENT

During the preparation of this work, we used generative AI tools to provide feedback on research methodology, assist with code implementation of the method, draft parts of the paper, and improve language readability. After using these tools, we reviewed and edited the content as needed and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

The model architecture and training objective are described in Section 3. The implementation details, including training hyperparameters, data processing, and the training protocol, are provided in Section 4.1. DASTrack is built on the publicly released DINOv3-L weights and trained on public datasets. All results are evaluated on public benchmarks under their standard protocols. Source code will be released upon acceptance.

REFERENCES

Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. Artrackv2: Prompting autoregressive tracker where to look and how to describe. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19048–19057, 2024.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19258–19267, 2024.

Wenrui Cai, Qingjie Liu, and Yunhong Wang. Spmtrack: Spatio-temporal parameter-efficient fine-tuning with mixture of experts for scalable visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16871–16881, 2025.

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*, pp. 9589–9600, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

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*, pp. 14572–14581, 2023.

Xin Chen, Ben Kang, Wanting Geng, Jiawen Zhu, Yi Liu, Dong Wang, and Huchuan Lu. Sutrack: Towards simple and unified single object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 2239–2247, 2025.

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*, pp. 13608–13618, 2022.

Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4655–4664, 2019.

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. In *Proceedings of the International Conference on Learning Representations*, pp. 1–22, 2021.

Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5374–5383, 2019.

Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, et al. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129(2):439–461, 2021.

Shenyuan Gao, Chunluan Zhou, Chao Ma, Xinggang Wang, and Junsong Yuan. Aiatrack: Attention in attention for transformer visual tracking. In *Proceedings of the European Conference on Computer Vision*, pp. 146–164, 2022.

Shenyuan Gao, Chunluan Zhou, and Jun Zhang. Generalized relation modeling for transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18686–18695, 2023.

Mingzhe Guo, Weiping Tan, Wenyu Ran, Liping Jing, and Zhipeng Zhang. Dreamtrack: Dreaming the future for multimodal visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7201–7210, 2025.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *Proceedings of the International Conference on Learning Representations*, pp. 1–26, 2022.

Kaijie He, Canlong Zhang, Sheng Xie, Zhixin Li, and Zhiwen Wang. Target-aware tracking with long-term context attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 773–780, 2023.

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*, pp. 16000–16009, 2022.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*, pp. 1–13, 2022.

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.

Yuqing Huang, Xin Li, Zikun Zhou, Yaowei Wang, Zhenyu He, and Ming-Hsuan Yang. Rtracker: Recoverable tracking via pn tree structured memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19038–19047, 2024.

Sajid Javed, Martin Danelljan, Fahad Shahbaz Khan, Muhammad Haris Khan, Michael Felsberg, and Jiri Matas. Visual object tracking with discriminative filters and siamese networks: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6552–6574, 2022.

Ben Kang, Xin Chen, Simiao Lai, Yang Liu, Yi Liu, and Dong Wang. Exploring enhanced contextual information for video-level object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4194–4202, 2025.

Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1125–1134, 2017.

Xiaohai Li, Bineng Zhong, Qihua Liang, Guorong Li, Zhiyi Mo, and Shuxiang Song. Mambalct: Boosting tracking via long-term context state space model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4986–4994, 2025.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Proceedings of the European Conference on Computer Vision*, pp. 280–296, 2022.

Liting Lin, Heng Fan, Zhipeng Zhang, Yaowei Wang, Yong Xu, and Haibin Ling. Tracking meets lora: Faster training, larger model, stronger performance. In *Proceedings of the European Conference on Computer Vision*, pp. 300–318, 2024.

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 *Proceedings of the European Conference on Computer Vision*, pp. 740–755, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, pp. 1–8, 2019.

Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *Proceedings of the European Conference on Computer Vision*, pp. 445–461, 2016.

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*, pp. 300–317, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Liang Peng, Junyuan Gao, Xinran Liu, Weihong Li, Shaohua Dong, Zhipeng Zhang, Heng Fan, and Libo Zhang. Vasttrack: Vast category visual object tracking. *Advances in Neural Information Processing Systems*, 37:130797–130818, 2024.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.

Liangtao Shi, Bineng Zhong, Qihua Liang, Ning Li, Shengping Zhang, and Xianxian Li. Explicit visual prompts for visual object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4838–4846, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *Transactions on Machine Learning Research*, 2026.

Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2321–2329, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Fast-itpn: Integrally pre-trained transformer pyramid network with token migration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9766–9779, 2024.

Narek Tumanyan, Assaf Singer, Shai Bagon, and Tali Dekel. Dino-tracker: Taming dino for self-supervised point tracking in a single video. In *Proceedings of the European Conference on Computer Vision*, pp. 367–385, 2024.

Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13763–13773, 2021.

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*, pp. 9697–9706, 2023.

Qiangqiang Wu, Tianyu Yang, Ziquan Liu, Baoyuan Wu, Ying Shan, and Antoni B. Chan. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14561–14571, 2023.

Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19300–19309, 2024.

Jinxia Xie, Bineng Zhong, Qihua Liang, Ning Li, Zhiyi Mo, and Shuxiang Song. Robust tracking via mamba-based context-aware token learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8727–8735, 2025.

Bin Yan, Xinyu Zhang, Dong Wang, Huchuan Lu, and Xiaoyun Yang. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5289–5298, 2021.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *Proceedings of the European Conference on Computer Vision*, pp. 341–357, 2022.

Jie Zhao, Xin Chen, Yongsheng Yuan, Michael Felsberg, Dong Wang, and Huchuan Lu. Efficient motion prompt learning for robust visual tracking. In *Proceedings of the International Conference on Machine Learning*, pp. 1–18, 2025.

Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7588–7596, 2024.

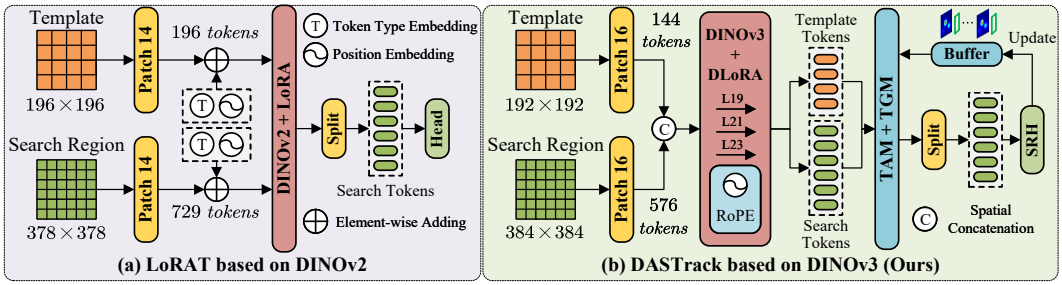

Figure A1: Comparison between LoRAT (Lin et al., 2024) and DASTrack. LoRAT adds position and token type embeddings to the template and search tokens separately, and predicts from the final-layer search tokens. DASTrack spatially concatenates the two inputs, applies RoPE within the backbone, and aggregates search tokens from multiple layers. The aggregated features are then modulated by TAM and TGM and refined by SRH, generating the final prediction.

A Supplementary Material

A.1 DASTrack based on DINOv3 vs. LoRAT based on DINOv2

LoRAT (Lin et al., 2024) is a representative tracker closely related to DASTrack, as both adapt a DINO-pretrained ViT-L backbone to tracking. Figure A1 compares our pipeline with theirs in four aspects, which are summarized as follows.

- **Position embedding.** LoRAT embeds the template and search region as two separate images with shared position embeddings, and therefore introduces additional token type embeddings to differentiate the two inputs. In contrast, DASTrack spatially concatenates them into a single image, where template and search tokens occupy distinct positions, allowing RoPE within the backbone to differentiate them without extra embeddings.
- **Number of tokens.** With a patch size of 16, DASTrack processes 720 tokens instead of 925, which accounts for its lower backbone computational cost in Table 4.
- **Backbone pre-training.** Compared with DINOv2 (Oquab et al., 2024), DINOv3 tackles the degradation of patch features during long training, where patches become similar to unrelated regions as the image-level objective gradually dominates. Specifically, DINOv3 introduces Gram anchoring, which anchors the pairwise patch similarities (i.e., the Gram matrix) to an earlier checkpoint with more consistent patch features, without hindering further improvement of the features. Moreover, the layer-wise evaluation of DINOv3 shows that performance on geometry-related tasks, such as depth estimation and tracking, peaks at an intermediate layer rather than the final one (Siméoni et al., 2026). This finding motivates DLoRA to aggregate multi-layer features.
- **Adaptive modulation and spatial refinement.** LoRAT predicts each frame independently with an MLP head. In contrast, DASTrack modulates the search features based on target appearance and tracking history, and refines the search features and bounding box iteratively.

A.2 Attribute Analysis on UAV123

Figure A2 presents the per-attribute success plots on UAV123, comparing DASTrack with LoRAT-L (Lin et al., 2024), SPMTrack-B (Cai et al., 2025), MCITrack-L (Kang et al., 2025), EVP-Track384 (Shi et al., 2024), HIPTrack (Cai et al., 2024), SeqTrack (Chen et al., 2023), OSTrack (Ye et al., 2022), and ROMTrack (Cai et al., 2023). DASTrack ranks first in three of the twelve attributes and second in seven. Notably, DASTrack achieves the highest score under similar object, which we attribute to the adaptive modulation in TAM and TGM that highlights the target and suppresses distractors. DASTrack also ranks first under scale variation and viewpoint change, indicating robustness to changes in target scale and appearance. However, DASTrack is relatively weaker under fast motion and out-of-view, ranking fourth and sixth, respectively. On LaSOT, DASTrack also lags behind LoRAT-L under fast motion, but surpasses it under out-of-view (Figure 6). We attribute the weakness under fast motion to large displacements, which may move away from both the search region and the historical trajectory location.

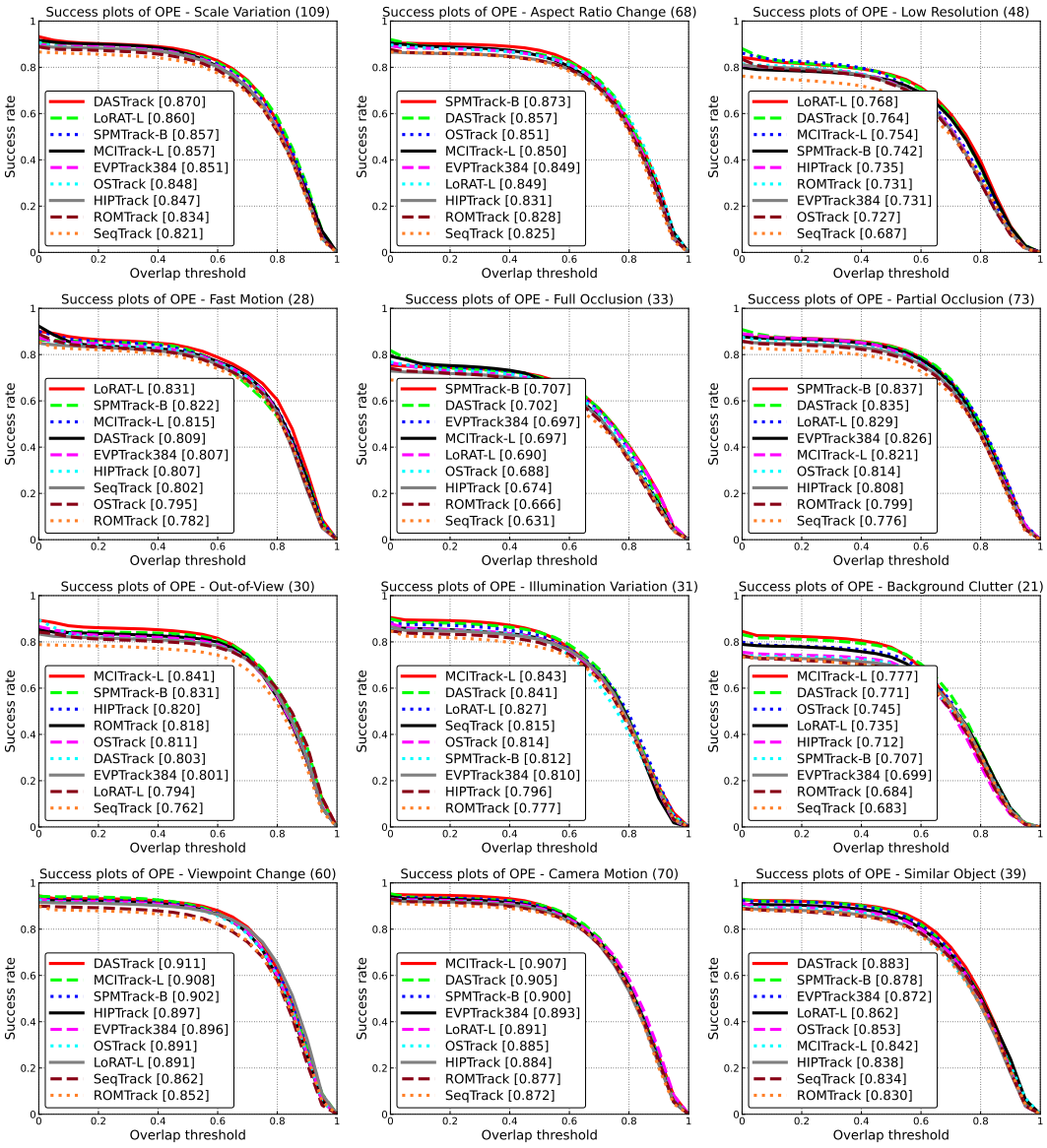

Figure A2: Per-attribute success plots on UAV123. The numbers in brackets denote $SR_{0.5}$ scores.

A.3 Qualitative Results on LaSOT

Similar Object. Figure A3 shows three sequences where visually similar distractors surround the target. Competing trackers frequently drift to nearby objects with similar appearance, while DASTrack maintains more stable tracking, which we attribute to the adaptive modulation in TAM and TGM that highlights the target and suppresses distractors.

Full Occlusion. Figure A4 shows three sequences where the target undergoes full occlusion and reappears. In all three sequences, DASTrack re-localizes the target shortly after reappearance, while competing trackers remain locked onto other objects or regions.

Fast Motion and Out-of-View. Figure A5 shows three sequences involving rapid displacement and target disappearance. In *yoyo-17*, DASTrack follows the fast-moving yoyo while most competing trackers drift to distractors. When the target leaves the view, the predicted box of DASTrack tends to expand. Since the search region is cropped in proportion to the predicted box, this expansion enlarges the search region in subsequent frames, which may help re-localize the target when it reappears.

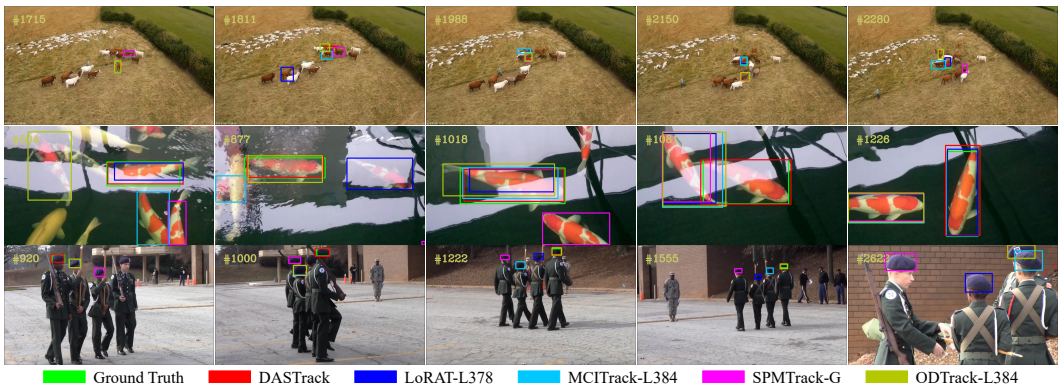

Figure A3: Qualitative comparison under similar object scenarios on LaSOT. From top to bottom: *cattle-13*, *goldfish-8*, and *hat-5*. Zoom in for better view.

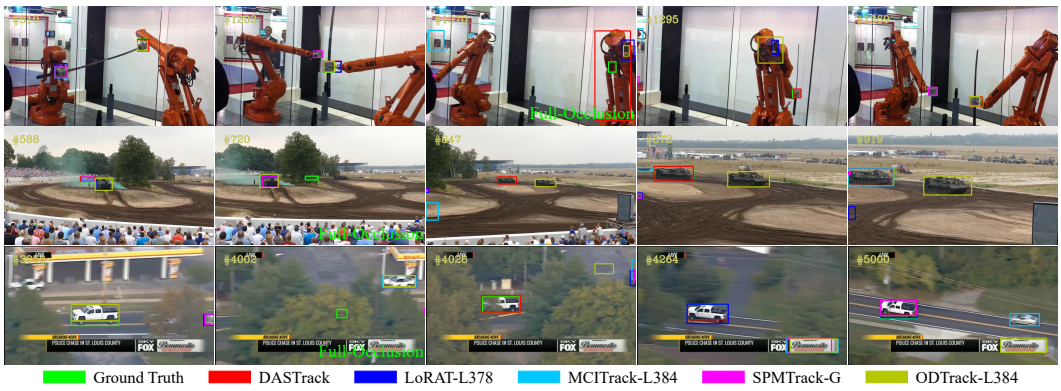

Figure A4: Qualitative comparison under full occlusion scenarios on LaSOT. From top to bottom: *robot-5*, *tank-9*, and *truck-16*. Zoom in for better view.

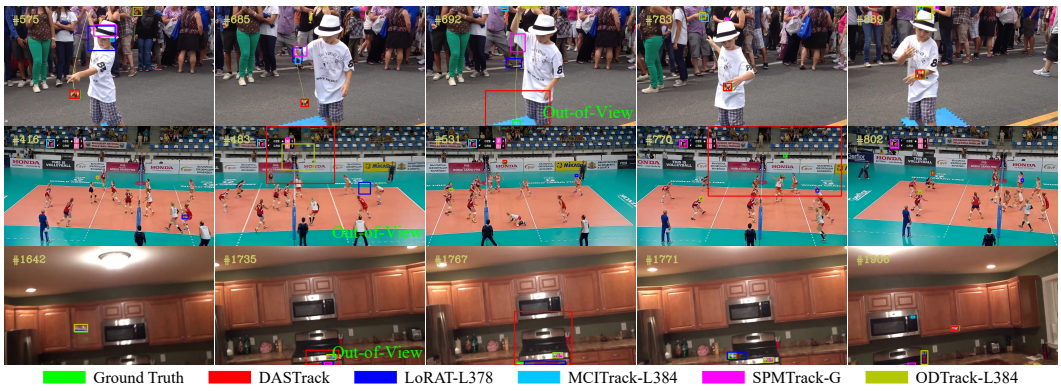

Figure A5: Qualitative comparison under fast motion and out-of-view scenarios on LaSOT. From top to bottom: *yoyo-17*, *volleyball-19*, and *drone-7*. Zoom in for better view.