# Instance-level Visual Active Tracking with Occlusion-Aware Planning

**Haowei Sun**[1*] **Kai Zhou**[1*] **Hao Gao**[1*] **Shiteng Zhang**[1] **Jinwu Hu**[1 2]
**Xutao Wen**[1] **Qixiang Ye**[4] **Mingkui Tan**[1 3†]

[1] South China University of Technology, [2] Pazhou Laboratory, [3] Key Laboratory of Big Data and Intelligent Robot, Ministry of Education,
[4] University of Chinese Academy of Sciences

## Abstract

*Visual Active Tracking (VAT) aims to control cameras to follow a target in 3D space, which is critical for applications like drone navigation and security surveillance. However, it faces two key bottlenecks in real-world deployment: confusion from visually similar distractors caused by insufficient instance-level discrimination and severe failure under occlusions due to the absence of active planning. To address these, we propose **OA-VAT**, a unified pipeline with three complementary modules. First, a training-free Instance-Aware Offline Prototype Initialization aggregates multi-view augmented features via DINOv3 to construct discriminative instance prototypes, mitigating distractor confusion. Second, an Online Prototype Enhancement Tracker enhances prototypes online and integrates a confidence-aware Kalman filter for stable tracking under appearance and motion changes. Third, an Occlusion-Aware Trajectory Planner, trained on our new `Planning-20k` dataset, uses conditional diffusion to generate obstacle-avoiding paths for occlusion recovery. Experiments demonstrate OA-VAT achieves 0.93 average SR on UnrealCV (+2.2% vs. SOTA TrackVLA), 90.8% average CAR on real-world datasets (+12.1% vs. SOTA GC-VAT), and 81.6% TSR on a DJI Tello drone. Running at 35 FPS on an RTX 3090, it delivers robust, real-time performance for practical deployment. The code is available at `https://github.com/SHWplus/OA-VAT`.*

## 1. Introduction

Visual Active Tracking (VAT) aims to dynamically control cameras to follow a specific target instance in 3D space [4, 12, 26, 39, 42, 51–55]. It is widely used in real-world applications such as navigation [16, 24, 43, 49] and security surveillance [14, 35, 46, 50]. Unlike passive visual tracking [1, 3, 5, 21, 38, 44, 47, 56] that operates on pre-recorded videos, VAT requires the agent to move the camera based on real-time visual observations. Passive visual tracking often falls short in the real world due to the dynamic nature of most targets. Thus, VAT offers a more practical yet challenging solution for real-world tracking applications.

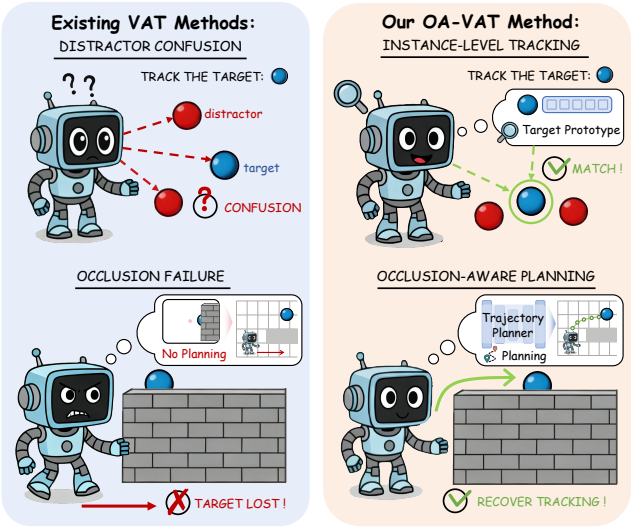
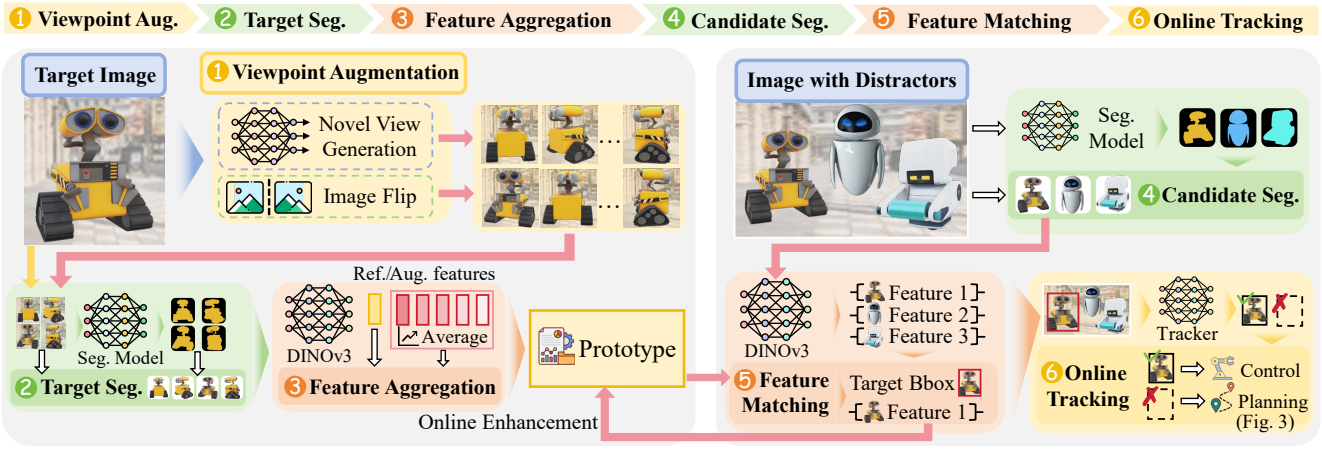



Figure 1. Comparison of existing VAT methods and OA-VAT. During offline initialization, we construct a target prototype to robustly match the target against distractors. During tracking, we employ a trajectory planner to recover the target under occlusions.

Existing VAT approaches are broadly categorized into reinforcement learning (RL)-based [12, 26, 39, 52, 55] and pipeline methods [4, 27, 42, 54]. RL-based methods learn end-to-end policies that map pixels to actions, enabling low-latency control without intermediate modules. However, they often struggle with sparse reward signals, leading to poor convergence in complex environments. Additionally, RL-based methods rely on simulation [11, 12, 32, 39] for policy training, which limits their deployment due to the sim-to-real gap. In contrast, pipeline methods decouple tracking into separate perception and control stages. These methods use pre-trained visual models [2, 6, 21, 22, 30, 41] for perception, enabling strong generalization to unseen en-

*Equal contribution. Email: sunhoward1105@gmail.com, kayjoe0723@gmail.com, hgao2729@gmail.com
†Corresponding author. Email: mingkuitan@scut.edu.cn




(a) Diagram of Instance-Aware Offline Prototype Initialization.

(b) Diagram of Online Prototype Enhancement Tracker.

Figure 2. Overview of Occlusion-Aware-VAT (OA-VAT). (a) Given a reference image, OA-VAT first constructs an instance-aware prototype offline by aggregating features from viewpoint augmentations. (b) At runtime, OA-VAT first detects the target via matching prototype and then tracks it online with prototype enhancement. When tracking fails, it activates the planning module (Fig. 3(b)) to recover target tracking.

vironments. Consequently, pipeline VAT methods offer a more deployable alternative under real-world constraints.

Unfortunately, existing pipeline methods remain challenging in real scenarios, partly for the following reasons (Fig. 1). **1) Lack of instance-level tracking ability.** Real-world tracking often involves multiple similar distractors. However, most existing VAT methods [4, 12, 27, 39, 55] operate at the **category level** and struggle to distinguish specific target instances. **2) Missing active occlusion handling.** Most pipeline VAT methods [10, 27, 31] employ simple controllers (*e.g.*, PID [28]) that only center the target in the image, without considering occlusions. While simplifying system design, such controllers cannot navigate trackers around obstacles to recover the target, often causing failure.

To address these, we propose a pipeline VAT method called OA-VAT (see Fig. 2), which enables instance-level discrimination and active recovery of occluded targets. First, we introduce an **Instance-Aware Offline Prototype Initialization** module. It extracts discriminative prototype features from the target's reference image. Second, we propose an **Online Prototype Enhancement Tracker** that tracks the target by matching the prototype to current frames. It uses online prototype enhancement to handle target appearance changes and a confidence-aware Kalman filter for motion prediction. Third, we develop an **Occlusion-Aware Trajectory Planner** to recover tracking during occlusions. Trained on our new `Planning-20k` dataset, it actively guides the camera around obstacles to restore target visibility. Our main contributions are as follows:

- **A Robust Instance-Level Tracker.** We propose a training-free tracker that initializes a discriminative prototype from a reference image, enhances it online, and integrates a confidence-aware Kalman filter for robust tracking under appearance and motion changes.
- **An Active Occlusion-Aware Planner.** We develop a path planner trained on our new `Planning-20k` dataset that plans collision-free trajectories to recover occluded targets and generalizes to arbitrary unseen targets.
- **Extensive Validation.** Experiments on simulators, real-world images, and a drone demonstrate OA-VAT's state-of-the-art performance and real-time inference.

## 2. Task Definition

Visual Active Tracking (VAT) aims to dynamically control a camera to follow a specific target instance in 3D space. The tracking agent must cope with challenges such as distractors, occlusions, and highly dynamic environments. We formulate VAT as a continuous visual control problem, characterized by the observation space, action space, and success criterion described below, and solved by a VAT tracker.

**Observation space $\mathcal{O}$.** At each time step, the agent receives an RGB image $\mathcal{I}_t$ (*e.g.*, $160 \times 120$ pixels). Besides, a single reference image $\mathcal{I}_{ref}$, depicting the target instance to be tracked, is provided at the beginning of the episode.

**Action space $\mathcal{A}$.** The agent operates in a continuous action space $\mathcal{A} \in \mathbb{R}^4$, with $a_t = [v_f, v_l, v_v, \omega_y]^T$ representing linear velocities (forward, lateral, vertical) and yaw rotation.

**VAT Tracker.** We define a VAT tracker as an embodied agent parameterized by a policy $\pi_\theta$, which maps visual observations to continuous control actions, i.e.,

$$a_t = \pi_\theta(\mathcal{I}_t, \mathcal{I}_{ref}). \tag{1}$$

**Success criterion.** We define a success criterion when the tracker keep the target instance centered in the image for a long duration. Metrics are detailed in Sec. 4.1.

## 3. Handling Distractors and Occlusions in VAT

Visual active tracking in the real world is extremely challenging, primarily due to the prevalence of distractors and frequent occlusions. To address these challenges, we propose Occlusion-Aware VAT (**OA-VAT**), an instance-level tracking method with occlusion awareness, aiming to improve visual active tracking in complex real-world applications. As shown in Fig. 2, OA-VAT incorporates a training-free instance tracking module that distinguishes the target from distractors. Moreover, we train a planning policy capable of recovering occluded targets. (see Fig. 3).

### 3.1. Instance-Aware Offline Prototype Initialization

We seek to build a training-free extraction module to obtain discriminative prototype features of the target instance. While existing visual foundation models (*e.g.*, Grounding-DINO [25], SAM [19], DINOv3 [37]) excel at category-level detection and segmentation, they struggle to provide features that are sufficiently discriminative at the instance level. Consequently, directly representing the target with their raw features or masks often leads to confusion with distractors. To address this, we propose an offline module that extracts a robust prototype from foundation model outputs without any additional training. Furthermore, we provide theoretical analysis for its effectiveness in Sec. 3.4.

Given a reference image $\mathcal{I}_{ref}$ of the target, we apply viewpoint augmentations and construct a prototype that captures view-invariant features for robust tracking. Specifically, we augment $\mathcal{I}_{ref}$ with horizontal and vertical flips to increase appearance diversity, producing an augmented image set $\{\mathcal{I}_i\}_i$. For human targets, whose appearance varies significantly across viewpoints, we generate additional views using an off-the-shelf diffusion model [48] and augment them with the same flips. We then employ a segmentation model $\texttt{Seg}(\cdot)$ (we use YOLO-E [40] for its speed-performance balance) to obtain the target mask and crop the instance in the reference and augmented images. This produces a set of target crops $\tilde{\mathbf{I}} = \{\tilde{\mathcal{I}}_{ref}, \tilde{\mathcal{I}}_1, ..., \tilde{\mathcal{I}}_N\}$, where $\tilde{\mathcal{I}}_i$ denotes the target crop of $\mathcal{I}_i$.

Subsequently, we employ a feature descriptor $\texttt{Desc}(\cdot)$, specifically DINOv3 [37] followed by global average pooling to extract features from the set $\tilde{\mathbf{I}}$:

$$\mathbf{f}_{ref} = \texttt{Desc}(\tilde{\mathcal{I}}_{ref}),\ \mathbf{f}_i = \texttt{Desc}(\tilde{\mathcal{I}}_i),\ i = 1, \dots, N, \quad (2)$$

where $\mathbf{f}_{ref}$ and $\mathbf{f}_i$ are the feature vectors for the reference and the $i$-th augmented image, respectively. The initial visual prototype $\tilde{\mathbf{f}}$ is obtained by averaging the original feature with the mean feature of the augmented set as follows:

$$\tilde{\mathbf{f}} = \frac{\mathbf{f}_{ref} + \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}_i}{\|\mathbf{f}_{ref} + \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}_i\|_2}. \quad (3)$$

The complete pipeline for the instance-aware offline prototype initialization module is shown in Fig. 2(a).

**Algorithm 1** Online Prototype Enhancement Tracker

**Require:** Frame $\mathcal{I}_t$, prototype $\tilde{\mathbf{f}}'$, bounding box $\mathbf{b}_t$, seg. model $\texttt{Seg}(\cdot)$, feature descriptor $\texttt{Desc}(\cdot)$, similarity threshold $\eta_s$, confidence threshold $\eta_c$, tracker $\mathcal{T}$
**Ensure:** Updated prototype $\tilde{\mathbf{f}}'$, predicted box $\mathbf{b}_{t+1}$

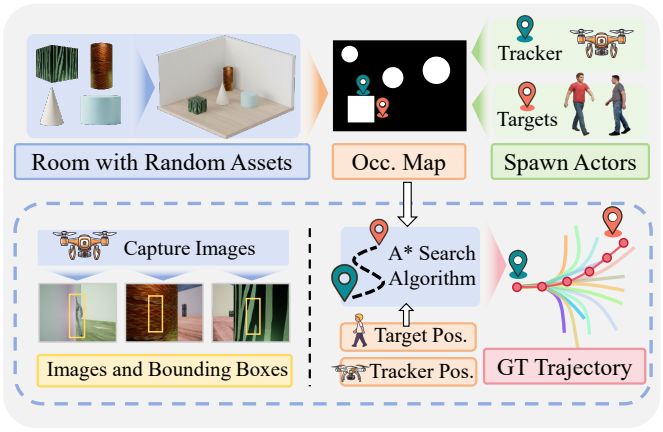
```
if b_t = ∅ then
    % Detection via Prototype Matching
    Obtain M candidate masks: {M^i}_{i=1}^M ← Seg(I_t)
    Crop candidates: I^i_cand ← I_t ⊙ M^i, i ∈ {1, ..., M}
    Extract features: f^i_cand ← Desc(I^i_cand)
    Compute cosine similarities S via Eq. (4)
    Find best candidate: i* ← arg max_i S_i
    b_{t+1} ← ∅
    if S_{i*} > η_s then
        b_{t+1} ← bounding box of I^{i*}_cand
    end if
else
    % Online Tracking with Prototype Enhancement
    Update tracker T with I_t and b_t
    Get confidence c_t and bounding box z_t from T
    % Confidence-Aware Kalman Filter
    Predict x̂_{t|t-1} via Eq. (6) and b_{t+1} ← H x̂_{t|t-1}
    if c_t < η_c then
        z_t ← ∅
    else
        Update state x̂_{t|t} via Eq. (7)
        Get predicted bounding box b_{t+1} ← H x̂_{t|t}
        Extract normalized target feature f̂_tar
        Enhance Prototype f̃' via Eq. (5)
    end if
end if
return f̃', b_{t+1}
```

### 3.2. Online Prototype Enhancement Tracker

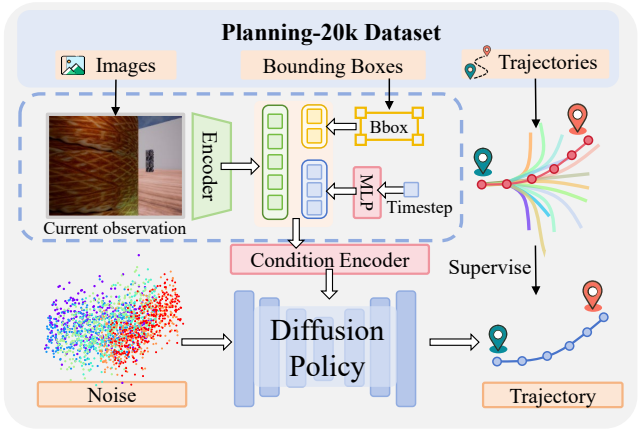
In the VAT setting, no initial bounding box is given, and the target's appearance and motion vary greatly during tracking. To address this, we first detect the target by matching the initialized prototype to the current frame. During tracking, we enhance the prototype online using new frames and predict the target's future motion to maintain robustness.

**Online Visual Prototype Enhancement**. Our enhancement strategy operates based on the current tracking state. If the tracker is uninitialized (i.e., the current bounding box $\mathbf{b}_t = \varnothing$), we start a target detection procedure. For each incoming observation frame $\mathcal{I}_t$, we first employ the segmentation model $\texttt{Seg}(\cdot)$ to extract $M$ candidate object masks belonging to the target category. These candidates are then cropped from the frame, producing a set of image crops $\mathcal{I}_{\text{cand}} = \{\mathcal{I}^1_{\text{cand}}, ..., \mathcal{I}^M_{\text{cand}}\}$. Each crop is then processed by $\texttt{Desc}(\cdot)$ to obtain feature $\mathbf{f}^i_{\text{cand}}$ for the $i$-th candidate.

We then compute the cosine similarity between current




(a) Diagram of the Planning-20k Data Collection Pipeline.




(b) Diagram of the Occlusion-Aware Planning Module.

Figure 3. Overview of the proposed Occlusion-Aware Trajectory Planner. The planner denoises a random trajectory into a feasible recovery path conditioned on image observations and a predicted target bounding box, enabling target-agnostic trajectory planning.

visual prototype $\tilde{\mathbf{f}}'$ and each candidate feature $\mathbf{f}^i_{\text{cand}}$ as:

$$S(\tilde{\mathbf{f}}', \mathbf{f}^i_{\text{cand}}) = \frac{\tilde{\mathbf{f}}' \cdot \mathbf{f}^i_{\text{cand}}}{\|\tilde{\mathbf{f}}'\|_2 \|\mathbf{f}^i_{\text{cand}}\|_2}, \tag{4}$$

where $\tilde{\mathbf{f}}'$ is initialized as $\tilde{\mathbf{f}}$. The candidate with the highest similarity that exceeds a predefined threshold $\eta_s$ is identified as the target instance. Once a candidate is matched, its bounding box and the current frame $\mathcal{I}_t$ are used to initialize an object tracker [45] $\mathcal{T}$ for precise localization.

During tracking, the tracker captures the target from varying viewpoints. These frames provide online features for enhancing the visual prototype. Specifically, we extract the normalized target feature $\hat{\mathbf{f}}_{\text{tar}}$ via $\texttt{Desc}(\cdot)$, and update the prototype via an exponential moving average (EMA):

$$\tilde{\mathbf{f}}' \leftarrow \beta \tilde{\mathbf{f}}' + (1-\beta)\hat{\mathbf{f}}_{\text{tar}}, \tag{5}$$

where $\beta$ is the EMA momentum. To avoid slowing down the online tracking, the prototype is updated in a separate thread. This strategy ensures the prototype continuously adapts to the appearance variations, maintaining discriminability over long tracking sequences.

**Confidence-Aware Kalman Filter.** To handle rapidly changing target motions, we design a confidence-aware Kalman filter that adapts the filter parameters based on the tracker's confidence to ensure robust motion prediction. We define the state vector as $\mathbf{x}_t = [x, y, w, h, \dot{x}, \dot{y}, \dot{w}, \dot{h}]^T$, representing the bounding box and its derivatives. The standard Kalman filter [18] operates via a predict-update cycle, where the state prediction $\hat{\mathbf{x}}_{t|t-1}$ is given by:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}\hat{\mathbf{x}}_{t-1|t-1}, \tag{6}$$

where $\mathbf{F}$ is the state transition matrix. Given the current confidence $c_t$ and bounding box $\mathbf{z}_t$ from tracker $\mathcal{T}$, the state is updated when $c_t$ exceeds the threshold $\eta_c$:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}\right), \tag{7}$$

where $\mathbf{H}$ is the observation matrix and $\mathbf{K}_t$ is the Kalman gain, given by:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^\top\left(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R}_t\right)^{-1}, \tag{8}$$

where $\mathbf{P}$ denotes the state covariance, and $\mathbf{R}_t$ is the measurement noise covariance. When the noise is large, the uncertainty in $\mathbf{z}_t$ increases, leading to a smaller $\mathbf{K}_t$, which assigns less weight to the current observation. The predicted bounding box can be obtained from $\hat{\mathbf{x}}_{t|t}$: $\mathbf{b}_{t+1} \leftarrow \mathbf{H}\hat{\mathbf{x}}_{t|t}$.

Our confidence-aware Kalman filter adapts the noise $\mathbf{R}_t$ according to the confidence $c_t$. We hypothesize that $c_t$ reflects the uncertainty of the current observation. Thus, the noise variance $\sigma^2(\cdot)$ is modeled as a function of $c_t$:

$$\mathbf{R}_t = \sigma^2(c_t)\mathbf{I}, \quad \sigma^2(c_t) = \frac{1}{1+e^{\lambda\cdot(c_t-\gamma)}}, \tag{9}$$

where $\lambda$ and $\gamma$ are hyperparameters. This sigmoid mapping reduces $\sigma^2$ and increases trust in the observation when $c_t$ exceeds $\gamma$, while enlarging $\sigma^2$ when $c_t$ is low. Consequently, at low confidence, the Kalman gain $\mathbf{K}_t$ is suppressed, making the filter rely more on its internal state prediction than on the noisy observation. This mechanism also allows the filter to continue predicting the target motion as in Eq. (6) during tracking failures, increasing the likelihood of re-acquiring the target. If no $\mathbf{z}_t$ is available for many consecutive frames, the planning module is triggered (see Sec. 3.3). The pseudocode is shown in Algorithm 1.

### 3.3. Occlusion-Aware Trajectory Planner

Another key challenge in open-world active tracking is frequent occlusion. Existing pipeline methods often use PID controllers [28] that steer the tracker with the image-plane distance between the target and the center. While effective for direct pursuit, these methods fail under occlusion, as they cannot plan trajectories to navigate around obstacles.

To address this, we propose a planning module that generates recovery trajectories via imitation learning. Since expert demonstrations under occlusion are unavailable, we introduce `Planning-20k`, a synthetic dataset collected in UnrealCV [32]. Using this data, we train a diffusion model to predict tracker trajectories conditioned on visual observations and the target bounding box. This design makes the policy inherently *target-agnostic*, enabling zero-shot generalization to arbitrary unseen targets.

**Data Collection.** The design of our `Planning-20k` dataset is central to learning a target-agnostic occlusion-recovery policy. As illustrated in Fig. 3(a), we generate data in UnrealCV's `SimpleRoom` environment as follows.

We first construct each map by randomly placing obstacles in an empty room and building a 2D occupancy map. To simulate occlusions, we randomly select an obstacle and spawn the target on one of its bounding box edges $e_i$. The tracker is then placed on an adjacent edge to $e_i$ to capture an RGB image and the target bounding box. Samples where the target is fully visible are discarded, ensuring only non-trivial planning scenarios. To enhance visual diversity, we apply domain randomization to illumination and obstacle textures using a texture dataset [8]. Expert trajectories are then obtained via A* search algorithm [17] on the occupancy map. The final dataset contains 20k samples, including 8k with default textures and 12k with randomized ones. Each sample consists of an RGB image, a target bounding box, and the corresponding expert trajectory.

**Data Diversity.** Our `Planning-20k` covers common occlusion structures in real-world scenarios. **(1)** Single-side occlusion blocks one side of the view, requiring the planner to go around it. **(2)** Double-side occlusion involves occlusions on both sides, leaving only a central gap for precise navigation. **(3)** Corridor-type occlusion includes obstacles on the left, right, and rear side, mimicking entry into a narrow hallway from a room.

**Occlusion-Aware Planning Module.** To achieve robust trajectory planning under occlusions, we propose a planning module inspired by Diffusion Policy [7]. We formulate the trajectory planning problem as a conditional denoising diffusion process. Previous diffusion policies [7] conditioned solely on visual observations often overfit the visual appearance of specific targets, limiting their generalization when the target changes. *In contrast*, OA-VAT incorporates target bounding boxes as an explicit condition. This guides the model to learn *target-agnostic* trajectory planning, thereby enabling robust generalization to unseen targets.

Formally, let $\mathbf{A}_t$ denotes trajectory points over a horizon $T_p$. We model the conditional distribution $p(\mathbf{A}_t|\mathcal{I}_t, \mathbf{b}_t)$ where $\mathcal{I}_t$ is the image and $\mathbf{b}_t$ is the target bounding box. The diffusion process starts from noise $\mathbf{A}_t^K \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoises for $K$ steps via:

$$\mathbf{A}_t^{k-1} = \alpha\left(\mathbf{A}_t^k - \phi\epsilon_\theta(\mathcal{I}_t, \mathbf{b}_t, \mathbf{A}_t^k, k)\right) + \mathcal{N}(0, \sigma^2\mathbf{I}), \quad (10)$$

where $\epsilon_\theta$ is a noise prediction network [34] that estimates the gradient of the action score function, and $\alpha, \phi, \sigma$ are noise scheduling function of iteration $k$ [29]. We use the mean squared error (MSE) to define the training objective:

$$\mathcal{L} = \mathbb{E}_{k, \mathbf{A}_t^0, \boldsymbol{\varepsilon}}\left[\|\boldsymbol{\varepsilon} - \epsilon_\theta(\mathcal{I}_t, \mathbf{b}_t, \mathbf{A}_t^0 + \boldsymbol{\varepsilon}, k)\|^2\right], \quad (11)$$

where $\epsilon$ is the random noise. $\mathcal{L}$ encourages the network to reconstruct the noise added to the ground-truth trajectory.

During online tracking, when the tracker (in Sec. 3.2) becomes unreliable, *i.e.*, $c_t < \eta_c$, we employ the confidence-aware Kalman filter in Eq. (6) to predict the bounding box $\mathbf{b}_t$ of the occluded target. $\mathbf{b}_t$ is then used as the conditioning input to our planner, generating a recovery trajectory.

**Why our planner generalizes well.** Our planner achieves robust generalization by modeling occlusions with physical rules. Specifically, it infers navigable pathways from spatial relationships between targets and obstacles. Moreover, we use bounding boxes as conditions, ensuring the planner focuses on target locations rather than visual textures. This captures general physical rules, and avoids reliance on large-scale photorealistic data.

### 3.4. Theoretical Guarantees on Instance Prototype

For any target instance $T_k$, we define its true feature manifold $M_k$ as the set of all features extracted by $\texttt{Desc}(\cdot)$ under arbitrary imaging conditions. From this manifold, the normalized reference features of $T_k$ are obtained via Eq. (2) and denoted by the set $F_k^*$. For each reference feature $f_k^* \in F_k^*$, we generate a corresponding set of multi-view augmented features $\{f_{k,i}\}_{i=1}^N$. The instance-aware prototypes are then derived via Eq. (3) and form the set $\hat{F}_k$. For any manifold $M$, $\mathbb{E}_{g\sim M}[\cdot]$ denotes expectation over $M$.

**Proposition 1** *Under the assumptions that (i) multi-view augmented features $\{f_{k,i}\}_{i=1}^N$ better cover the true feature manifold $M_k$ than the reference feature $f_k^*$ and (ii) features from the same target are cohesive while those from different targets are well-separated, for any two distinct targets $T_k \neq T_j$, the minimum squared distance between any pair of $(\hat{f}_k, \hat{f}_j)$ sampled from $\hat{F}_k$ and $\hat{F}_j$ is larger than that between any pair of $(f_k^*, f_j^*)$ sampled from $F_k^*$ and $F_j^*$:*

$$\min_{\hat{f}_k\in\hat{F}_k, \hat{f}_j\in\hat{F}_j} \|\hat{f}_k - \hat{f}_j\|_2^2 \geq \min_{f_k^*\in F_k^*, f_j^*\in F_j^*} \|f_k^* - f_j^*\|_2^2. \quad (12)$$

*For the proof of Proposition 1, please refer to the Appendix.* Proposition 1 shows that the instance-aware offline prototype initialization module (Sec. 3.1) improves inter-instance separation by aggregating multi-view features, producing prototypes more discriminative than the original foundation model features, as validated by Fig. 4.

Table 1. Results in UnrealCV environments containing **distractors**. **Bold** represents the best while underline represents the second. TrackVLA [42] does not report $AR$ and its training code and checkpoints are not publicly available, so we cannot reproduce this metric.

| Tracker | Publication | Parking Lot (2D) | | | UrbanCity (4D) | | | ComplexRoom (4D) | | | Average | | | Params. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $AR\uparrow$ | $EL\uparrow$ | $SR\uparrow$ | $AR\uparrow$ | $EL\uparrow$ | $SR\uparrow$ | $AR\uparrow$ | $EL\uparrow$ | $SR\uparrow$ | $AR\uparrow$ | $EL\uparrow$ | $SR\uparrow$ | |
| DiMP [4] | ICCV 2019 | 111 | 271 | 0.24 | 170 | 348 | 0.32 | 97 | 307 | 0.26 | 126 | 309 | 0.27 | $26M$ |
| SARL [26] | TPAMI 2019 | 53 | 237 | 0.12 | 74 | 221 | 0.16 | 22 | 263 | 0.15 | 50 | 240 | 0.14 | $2M$ |
| AD-VAT [51] | ICLR 2019 | 43 | 232 | 0.13 | 32 | 204 | 0.06 | 16 | 223 | 0.16 | 30 | 220 | 0.12 | $4M$ |
| AD-VAT+ [52] | TPAMI 2019 | 35 | 166 | 0.08 | 89 | 245 | 0.11 | 35 | 262 | 0.18 | 53 | 224 | 0.12 | $4M$ |
| TS [53] | ICML 2021 | 186 | 331 | 0.39 | 227 | 381 | 0.51 | 250 | 401 | 0.54 | 221 | 371 | 0.48 | $7M$ |
| EVT [55] | ECCV 2024 | 192 | 425 | 0.63 | 272 | 472 | 0.92 | 354 | 479 | 0.88 | 273 | 459 | 0.81 | $748M$ |
| FAn [27] | RAL 2024 | 126 | 301 | 0.28 | 167 | 334 | 0.30 | 189 | 351 | 0.29 | 161 | 329 | 0.29 | $132M$ |
| FAn+SAM2 [33] | ICLR 2025 | 170 | 349 | 0.40 | 201 | 407 | 0.55 | 262 | 422 | 0.61 | 211 | 393 | 0.52 | $122M$ |
| TrackVLA [42] | CoRL 2025 | - | 467 | 0.90 | - | 476 | 0.92 | - | 479 | 0.91 | - | 474 | 0.91 | $> 7B$ |
| **Ours** | CVPR 2026 | **392** | **482** | **0.93** | **385** | **486** | **0.95** | **392** | **481** | **0.92** | **390** | **483** | **0.93** | $584M$ |

Table 2. Performance on DAT benchmark under within scene setting.

| Tracker | Publication | citystreet | | desert | | village | |
|---|---|---|---|---|---|---|---|
| | | $CR\uparrow$ | $TSR\uparrow$ | $CR\uparrow$ | $TSR\uparrow$ | $CR\uparrow$ | $TSR\uparrow$ |
| SARL [26] | TPAMI 2019 | $49_{\pm 3}$ | $0.25_{\pm 0.02}$ | $9_{\pm 1}$ | $0.06_{\pm 0.00}$ | $46_{\pm 5}$ | $0.23_{\pm 0.03}$ |
| D-VAT [12] | RAL 2024 | $48_{\pm 8}$ | $0.26_{\pm 0.02}$ | $47_{\pm 13}$ | $0.26_{\pm 0.04}$ | $44_{\pm 8}$ | $0.22_{\pm 0.05}$ |
| GC-VAT [39] | NeurIPS 2025 | $279_{\pm 110}$ | $0.80_{\pm 0.30}$ | $307_{\pm 124}$ | $\mathbf{0.84}_{\pm 0.29}$ | $239_{\pm 134}$ | $0.73_{\pm 0.32}$ |
| **Ours** | CVPR 2026 | $\mathbf{310}_{\pm 2}$ | $\mathbf{0.83}_{\pm 0.01}$ | $\mathbf{311}_{\pm 3}$ | $0.83_{\pm 0.01}$ | $\mathbf{307}_{\pm 2}$ | $\mathbf{0.83}_{\pm 0.01}$ |
| Tracker | Publication | downtown | | lake | | farmland | |
| | | $CR\uparrow$ | $TSR\uparrow$ | $CR\uparrow$ | $TSR\uparrow$ | $CR\uparrow$ | $TSR\uparrow$ |
| SARL [26] | TPAMI 2019 | $54_{\pm 5}$ | $0.29_{\pm 0.01}$ | $47_{\pm 3}$ | $0.24_{\pm 0.02}$ | $60_{\pm 25}$ | $0.23_{\pm 0.01}$ |
| D-VAT [12] | RAL 2024 | $9_{\pm 1}$ | $0.06_{\pm 0.01}$ | $46_{\pm 8}$ | $0.26_{\pm 0.06}$ | $13_{\pm 1}$ | $0.07_{\pm 0.00}$ |
| GC-VAT [39] | NeurIPS 2025 | $203_{\pm 119}$ | $0.65_{\pm 0.30}$ | $181_{\pm 116}$ | $0.61_{\pm 0.31}$ | $243_{\pm 117}$ | $0.68_{\pm 0.32}$ |
| **Ours** | CVPR 2026 | $\mathbf{370}_{\pm 3}$ | $\mathbf{0.99}_{\pm 0.01}$ | $\mathbf{318}_{\pm 10}$ | $\mathbf{0.84}_{\pm 0.02}$ | $\mathbf{310}_{\pm 3}$ | $\mathbf{0.83}_{\pm 0.02}$ |

Table 3. Average results of VAT trackers in UnrealCV (details in Appendix).

| Tracker | Average | | |
|---|---|---|---|
| | $AR\uparrow$ | $EL\uparrow$ | $SR\uparrow$ |
| DiMP [4] | 204 | 367 | 0.58 |
| SARL [26] | 240 | 394 | 0.57 |
| AD-VAT [51] | 238 | 416 | 0.62 |
| AD-VAT+ [52] | 307 | 454 | 0.76 |
| TS [53] | 312 | 474 | 0.86 |
| RSPT [54] | 329 | 478 | 0.92 |
| EVT [55] | 297 | 490 | 0.95 |
| FAn [27] | 237 | 462 | 0.90 |
| FAn+SAM2 [33] | 257 | 474 | 0.94 |
| TrackVLA [42] | - | **500** | **1.00** |
| **Ours** | **391** | **500** | **1.00** |

# 4. Experiment

## 4.1. Experimental Settings

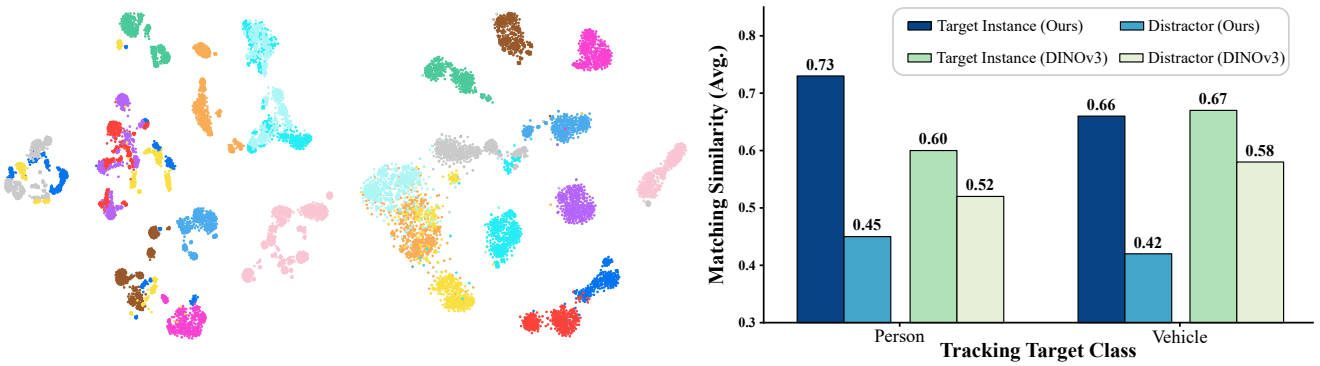
**Experimental Setup.** We evaluate OA-VAT in a zero-shot setting on two state-of-the-art VAT benchmarks: UnrealCV [32] and DAT [39]. In UnrealCV, we test OA-VAT on 3 challenging maps with distractors (Parking Lot (2D), UrbanCity (4D), ComplexRoom (4D)) and 5 single-target maps. In DAT, we perform evaluations under daytime condition across all six scenes, comparing OA-VAT with within-scene trained models. We further validate OA-VAT on real-world image datasets including VOT [20], DTB70 [23], and UAVDT [13], and deploy it on a *DJI Tello* drone [9] for real-world evaluation. *Implementation details and hyperparameter analysis are provided in the Appendix.*

**Metrics.** In UnrealCV, we use three metrics: *Accumulated Reward (AR)* measures the average reward over 100 episodes. *Episode Length (EL)* denotes the average steps per episode, with early termination if the target remains out of view for more than 50 consecutive steps. *Success Rate (SR)* represents the proportion of success episodes where the tracker successfully follows the target for 500 steps. In DAT, *Cumulative Reward (CR)* measures the average total reward over 40 episodes, and *Tracking Success Rate (TSR)* denotes the success rate within 1,500 steps.

**Baselines.** We compare OA-VAT against 12 baselines, grouped into two categories: the RL-based trackers include SARL [26], AD-VAT [51], AD-VAT+ [52], TS [53], D-VAT [12], GC-VAT [39], and EVT [55]. The remaining methods follow a pipeline design: DiMP [4], RSPT [54], Follow Anything (FAn) [27], FAn+SAM2 [33], and TrackVLA [42]. *Related works are detailed in the Appendix.*

## 4.2. Comparison Experiments

**Effectiveness and Efficiency in UnrealCV Environments with Distractors.** We evaluate OA-VAT in UnrealCV [32] maps with similar distractors. As shown in Tab. 1, OA-VAT achieves the best performance among all compared methods, including the SOTA method TrackVLA [42]. Specifically, OA-VAT outperforms TrackVLA by 2.2% in average success rate while requiring far less computation. We train OA-VAT on single RTX 3090 GPU for 15 hours, compared to TrackVLA's 24×H100 GPUs for the same duration. Furthermore, OA-VAT has a model size of 584M, significantly




(a) t-SNE on prototypes of DINOv3. (b) t-SNE on prototypes of OA-VAT. (c) Average cosine similarity of prototype to target and to distractors.

Figure 4. Ablation study on the instance-aware offline prototype initialization module. In (a) and (b), each point represents a prototype feature extracted from single view of an instance, with all points from the same instance assigned the same color. (c) shows the similarity of the offline initialized prototype with the target instance versus distractors during online tracking.

Table 4. Results of ablation experiments on UnrealCV environments containing distractors.

| Module | Setting | Parking Lot (2D) | | | UrbanCity (4D) | | | ComplexRoom (4D) | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *AR* | *EL* | *SR* | *AR* | *EL* | *SR* | *AR* | *EL* | *SR* | *AR* | *EL* | *SR* |
| Instance-Aware Offline Prototype Initialization | w/ DINOv3 [37] Prototype | 384 | 481 | 0.92 | 369 | 443 | 0.84 | 370 | 454 | 0.85 | 374 | 459 | 0.87 |
| | **Ours** | **392** | **482** | **0.93** | **385** | **486** | **0.95** | **392** | **481** | **0.92** | **390** | **483** | **0.93** |
| Online Visual Prototype Enhancement | w/o Online Enhancement | 352 | 463 | 0.84 | 340 | 469 | 0.77 | 354 | 477 | 0.84 | 349 | 470 | 0.82 |
| | w/ Average Enhancement | 356 | **485** | 0.90 | 347 | 477 | 0.88 | 359 | **481** | 0.89 | 354 | 481 | 0.89 |
| | **Ours** | **392** | 482 | **0.93** | **385** | **486** | **0.95** | **392** | **481** | **0.92** | **390** | **483** | **0.93** |
| Confidence-Aware Kalman Filter | w/o Kalman Filter | 349 | 474 | 0.87 | 343 | 459 | 0.88 | 334 | 459 | 0.85 | 342 | 464 | 0.87 |
| | w/ Linear Kalman Filter [18] | 356 | **488** | 0.88 | 359 | 457 | 0.91 | 346 | 463 | 0.90 | 354 | 469 | 0.90 |
| | **Ours** | **392** | 482 | **0.93** | **385** | **486** | **0.95** | **392** | **481** | **0.92** | **390** | **483** | **0.93** |
| Occlusion-Aware Trajectory Planner | w/o Planning (*i.e.*, PID [28]) | 345 | 478 | 0.83 | 360 | 470 | 0.88 | 353 | 460 | 0.84 | 353 | 469 | 0.85 |
| | w/ EVT Planning [55] | 308 | 472 | 0.82 | 303 | 475 | 0.88 | 360 | 482 | 0.90 | 324 | 476 | 0.87 |
| | w/o Bounding Box | 368 | 476 | 0.90 | 347 | 456 | 0.86 | **396** | **484** | 0.91 | 370 | 472 | 0.89 |
| | **Ours** | **392** | **482** | **0.93** | **385** | **486** | **0.95** | 392 | 481 | **0.92** | **390** | **483** | **0.93** |

smaller than both TrackVLA and EVT. It runs at **35 FPS** on an RTX 3090 GPU, notably faster than TrackVLA's 10 FPS on an RTX 4090 GPU, ensuring real-time performance.

Moreover, we also evaluate OA-VAT on distractor-free scenes. As shown in Tab. 3, OA-VAT achieves perfect performance ($SR = 1.00$) across all five scenes, maintaining the target in view for the full episode ($EL = 500$).

**Effectiveness in DAT Environments.** As shown in Tab. 2, OA-VAT consistently outperforms existing methods. It achieves average improvements of 32.6% in CR (242 $\rightarrow$ 321) and 19.4% in TSR (0.72 $\rightarrow$ 0.86) over GC-VAT [39]. The standard deviation of TSR is less than 0.02, indicating stable tracking performance. Notably, DAT targets are *vehicles*, which are unseen during training, demonstrating the strong zero-shot generalization of OA-VAT.

### 4.3. Ablation Experiments

We validate our prototype initialization module (Sec. 3.1) on video datasets [15, 36] through qualitative and quantitative analysis. We then conduct ablation studies on three UnrealCV maps with distractors to evaluate the effectiveness of our prototype initialization, prototype enhancement, confidence-aware Kalman filter, and the planning module.

**Effectiveness of Offline Prototype Initialization.** We compare our prototype (Sec. 3.1) with raw DINOv3 [37] features on *video 40* in PersonPath22 [36]. We extract a prototype for each target per frame. As shown in Fig. 4(a)-(b), our prototypes are well-separated across instances, while DINOv3 features largely overlap. Quantitatively, we compute the average cosine similarity of prototype to targets and to distractors on *video 40* (for persons) and all *car* videos in LaSOT [15] (for vehicles). OA-VAT achieves a similarity margin of 0.28 between person targets and distractors, greatly larger than DINOv3's 0.08 (Fig. 4(c)). When applied to tracking, OA-VAT improves avg. SR by 6.9% relative to the DINOv3 baseline, as shown in Tab. 4(rows 1-2).

**Effectiveness of Online Prototype Enhancement.** We compare our EMA-based enhancement against average-

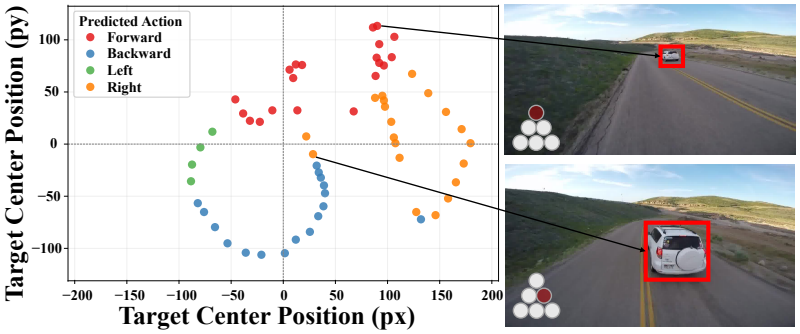



Figure 5. Results on real-world images of the *Car8* video in DTB70 [23] dataset.

Table 5. Effectiveness of OA-VAT on real-world image evaluation. We select eight videos from each of VOT [20], DTB70 [23] and UAVDT[13] datasets.

| Tracker | Average Correct Action Rate | | |
|---|---|---|---|
| | VOT [20] | DTB70 [23] | UAVDT [13] |
| Random | 0.413 | 0.426 | 0.421 |
| ORTrack [45] | 0.661 | 0.781 | 0.879 |
| ORTrack (w/o bbox) | 0.260 | 0.380 | 0.296 |
| FAn [27] | 0.720 | 0.719 | 0.592 |
| GC-VAT [39] | 0.795 | 0.833 | 0.802 |
| **Ours** | **0.879** | **0.900** | **0.945** |

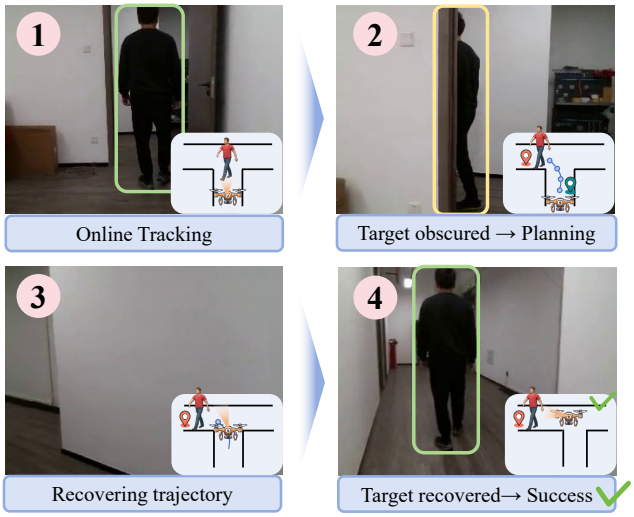
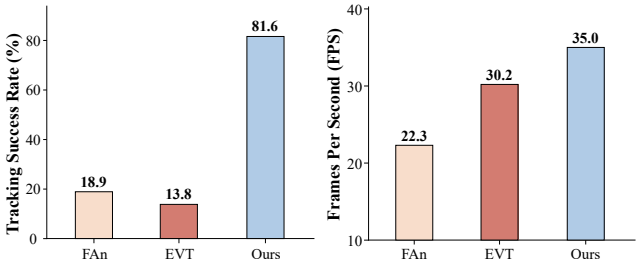



Figure 6. Recovery of prolonged occlusion on a *DJI Tello* drone.




(a) Success rate on real drones. (b) FPS on real drones.

Figure 7. Experiment results on real drone *DJI Tello*.

update and no-update variants. As shown in Tab. 4 (rows 3-5), it outperforms both no-update (+13.4% avg. SR) and average-update variant (+4.5% avg. SR), effectively balancing historical knowledge and recent observations.

**Effectiveness of Confidence-Aware Kalman Filter.** As shown in Tab. 4 (rows 6-8), OA-VAT improves avg. SR by 6.9% over the no-filter baseline, and outperforms the linear Kalman filter variant (+3.3% avg. SR), showing that state estimation helps correct the unreliable bounding boxes.

**Effectiveness of Planning Module.** We compare our planner with three alternatives: no planning (PID [28]), EVT planning [55] module based on offline RL, and a planner trained without bounding boxes as input. As shown in Tab. 4 (rows 9-12), OA-VAT achieves an average SR of 0.93, outperforming EVT by 6.9% and the variant without bounding box input by 4.5%. This confirms that bounding box guidance enhances planning robustness, and our OA-VAT can effectively recover trajectories of occluded targets.

### 4.4. Experiments in Real-world Scenarios

**Effectiveness on real-world images.** To assess OA-VAT 's transferability to real-world scenarios, we follow the setting of [39] and perform zero-shot evaluation on 8 videos each from VOT [20], DTB70 [23], and UAVDT [13]. Although camera control is unavailable in these videos, we can feed frames into the model and verify whether the predicted action would move the target toward the image center.

Qualitative results on video *Car8* from DTB70 dataset are shown in Fig. 5. Each point denotes the target's location in the image, with its color indicating the predicted action, and arrows showing visual observations. When the target deviates from the image center, OA-VAT correctly predicts actions to steer it back (e.g., OA-VAT outputs a rightward control when the target is on the right). Quantitatively, we use Correct Action Rate (CAR), i.e., the action-prediction accuracy, to evaluate the performance. As shown in Tab. 5, OA-VAT achieves an avg. CAR of 90.8%, outperforming GC-VAT [39] by 12.1%. *See Appendix for more results.*

Furthermore, to comprehensively evaluate robustness, we compare OA-VAT with ORTrack [45], the base tracker of the Online Prototype Enhancement Tracker in Sec. 3.2. It is important to note that ORTrack is a passive visual tracking model that necessitates an initial bounding box as input. As shown in Tab. 5, ORTrack lags behind our method by 13.4% on average (0.774 *vs.* 0.908), even though it uses ground-truth boxes for initialization. Moreover, ORTrack collapses when initialized with only a reference image.

**Effectiveness on real robots.** To assess the real-world applicability of OA-VAT beyond simulation and image-based benchmarks, we deploy it on a *DJI Tello* drone [9]. During operation, the drone transmits H.264 video streams over Wi-Fi to a ground station equipped with an NVIDIA RTX 3090 GPU. The station then decodes the video, runs OA-VAT to generate control commands, and sends them back to the drone for closed-loop tracking.

We evaluate OA-VAT in two challenging settings. In scenarios with distractors, OA-VAT accurately locates and re-identifies the designated target after temporary occlusions, demonstrating robust instance-level performance. Under prolonged occlusion, OA-VAT actively navigates around obstacles to recover the target (Fig. 6), whereas baseline methods such as EVT [55] and FAn [27] lose the target permanently. Moreover, We adopt Tracking Success Rate (TSR) to quantify tracking performance. As shown in Fig. 7(a), OA-VAT achieves a TSR of 81.6%, far surpassing the best baseline (18.9%). Furthermore, as shown in Fig. 7(b), OA-VAT runs at 35.0 FPS, enabling real-time operation in dynamic scenes. *See Appendix for more results.*

## 5. Conclusion

In this paper, we propose a pipeline VAT method, called OA-VAT. Specifically, we introduce an instance-aware offline prototype initialization module to mitigate distractor confusion. Then we propose an online tracker that enhances prototypes online and integrates a confidence-aware Kalman filter for stable tracking. Moreover, we develop a planner trained on our new `Planning-20k` dataset, which plans trajectories to recover occluded targets and generalizes to unseen targets. Experiments on simulators, real-world images, and a drone demonstrate OA-VAT's state-of-the-art performance and real-time inference.

## Acknowledgements

This work was partially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U24A20327).

## References

[1] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pages 983–990. IEEE, 2009. 1

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016. 1

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 1

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6182–6191, 2019. 1, 2, 6

[5] 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 (CVPR)*, pages 14572–14581, 2023. 1

[6] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024. 5

[8] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[9] Da-Jiang Innovations. Dji tello. https://store.dji.com/product/tello, 2025. 6, 9

[10] Dibyendu Kumar Das, Mouli Laha, Somajyoti Majumder, and Dipnarayan Ray. Stable and consistent object tracking: An active vision approach. In *Advanced Computational and Communication Paradigms: Proceedings of International Conference on ICACCP 2017, Volume 2*, pages 299–308. Springer, 2018. 2

[11] Alessandro Devo, Alberto Dionigi, and Gabriele Costante. Enhancing continuous control of mobile robots for end-to-end visual active tracking. *Robotics and Autonomous Systems*, 142:103799, 2021. 1

[12] Alberto Dionigi, Simone Felicioni, Mirko Leomanni, and Gabriele Costante. D-vat: End-to-end visual active tracking for micro aerial vehicles. *IEEE Robotics and Automation Letters*, 9(6):5046–5053, 2024. 1, 2, 6

[13] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018. 6, 8

[14] Bara J Emran and Homayoun Najjaran. A review of quadrotor: An underactuated mechanical system. *Annual Reviews in Control*, 46:165–180, 2018. 1

[15] 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*, pages 5374–5383, 2019. 7

[16] Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational*

*Linguistics: ACL 2023*, pages 3043–3061, Toronto, Canada, 2023. Association for Computational Linguistics. 1

[17] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 5

[18] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 4, 7

[19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 3

[20] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2137–2155, 2016. 6, 8

[21] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018. 1

[22] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4282–4291, 2019. 1

[23] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *Proceedings of the AAAI conference on artificial intelligence*, 2017. 6, 8

[24] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15384–15394, 2023. 1

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 3

[26] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking and its real-world deployment via reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1317–1332, 2019. 1, 6

[27] Alaa Maalouf, Ninad Jadhav, Krishna Murthy Jatavallabhula, Makram Chahine, Daniel M Vogt, Robert J Wood, Antonio Torralba, and Daniela Rus. Follow anything: Open-set detection, tracking, and following in real-time. *IEEE Robotics and Automation Letters*, 9(4):3283–3290, 2024. 1, 2, 6, 8, 9

[28] Nicolas Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922. 2, 4, 7, 8

[29] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 5

[30] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 1

[31] Neng Pan, Ruibin Zhang, Tiankai Yang, Can Cui, Chao Xu, and Fei Gao. Fast-tracker 2.0: Improving autonomy of aerial tracking with active vision and human location regression. *IET Cyber-Systems and Robotics*, 3(4):292–301, 2021. 2

[32] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1221–1224, 2017. 1, 5, 6

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 6

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[35] David C. Schedl, Indrajit Kurmi, and Oliver Bimber. An autonomous drone for search and rescue in forests using airborne optical sectioning. *Science Robotics*, 6, 2021. 1

[36] Bing Shuai, Alessandro Bergamo, Uta Buechler, Andrew Berneshawi, Alyssa Boden, and Joe Tighe. Large scale real-world multi person tracking. In *European Conference on Computer Vision*. Springer, 2022. 7

[37] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 3, 7

[38] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1442–1468, 2013. 1

[39] Haowei Sun, Jinwu Hu, Zhirui Zhang, Haoyuan Tian, Xinze Xie, Yufeng Wang, Xiaohua Xie, Yun Lin, Zhuliang Yu, and Mingkui Tan. Open-world drone active tracking with goal-centered rewards. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 1, 2, 6, 7, 8

[40] Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24591–24602, 2025. 3

[41] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019. 1

[42] Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. Trackvla: Embodied visual tracking in the wild. *arXiv preprint arXiv:2505.23189*, 2025. 1, 6

[43] Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology, 2024. 1

[44] Longyin Wen, Dawei Du, Pengfei Zhu, Qinghua Hu, Qilong Wang, Liefeng Bo, and Siwei Lyu. Detection, tracking, and counting meets drones in crowds: A benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7812–7821, 2021. 1

[45] You Wu, Xucheng Wang, Xiangyang Yang, Mengyuan Liu, Dan Zeng, Hengzhou Ye, and Shuiwang Li. Learning occlusion-robust vision transformers for real-time uav tracking. In *CVPR*, 2025. 4, 8

[46] Linjie Xing, Xiaoyan Fan, Yaxin Dong, Zenghui Xiong, Lin Xing, Yang Yang, Haicheng Bai, and Chengjiang Zhou. Multi-uav cooperative system for search and rescue based on yolov5. *International Journal of Disaster Risk Reduction*, 76:102972, 2022. 1

[47] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, pages 341–357. Springer, 2022. 1

[48] Chao Yuan, Guiwei Zhang, Changxiao Ma, Tianyi Zhang, and Guanglin Niu. From poses to identity: Training-free person re-identification via feature centralization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24409–24418, 2025. 3

[49] Hui Yuan, Yan Huang, Naigong Yu, Dongbo Zhang, Zetao Du, Ziqi Liu, and Kun Zhang. Multimodal pretrained knowledge for real-world object navigation. *Machine Intelligence Research*, 22(4):713–729, 2025. 1

[50] Chaoqun Zhang, Wenjuan Zhou, Weidong Qin, and Weidong Tang. A novel uav path planning approach: Heuristic crossing search and rescue optimization algorithm. *Expert Systems with Applications*, 215:119243, 2023. 1

[51] Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat: An asymmetric dueling mechanism for learning visual active tracking. In *International Conference on Learning Representations*, 2019. 1, 6

[52] Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1467–1482, 2019. 1, 6

[53] Fangwei Zhong, Peng Sun, Wenhan Luo, Tingyun Yan, and Yizhou Wang. Towards distraction-robust active visual tracking. In *International Conference on Machine Learning*, pages 12782–12792. PMLR, 2021. 6

[54] Fangwei Zhong, Xiao Bi, Yudi Zhang, Wei Zhang, and Yizhou Wang. Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3705–3714, 2023. 1, 6

[55] Fangwei Zhong, Kui Wu, Hai Ci, Churan Wang, and Hao Chen. Empowering embodied visual tracking with visual foundation models and offline rl. In *European Conference on Computer Vision*, pages 139–155. Springer, 2024. 1, 2, 6, 7, 8, 9

[56] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2022. 1