Open-World Drone Active Tracking with Goal-Centered Rewards

Haowei Sun[1 2*] **Jinwu Hu**[1 3*] **Zhirui Zhang**[1 2] **Haoyuan Tian**[1 2] **Xinze Xie**[1 2]
Yufeng Wang[1 5] **Xiaohua Xie**[6] **Yun Lin**[7] **Zhuliang Yu**[1 2†] **Mingkui Tan**[1 4†]
[1] South China University of Technology, [2] Institute for Super Robotics (Huangpu), [3] Pazhou Laboratory,
[4] Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, [5] Peng Cheng Laboratory,
[6] Sun Yat-sen University, [7] Harbin Engineering University

Abstract

Drone Visual Active Tracking aims to autonomously follow a target object by controlling the motion system based on visual observations, providing a more practical solution for effective tracking in dynamic environments. However, accurate Drone Visual Active Tracking using reinforcement learning remains challenging due to the absence of a unified benchmark and the complexity of open-world environments with frequent interference. To address these issues, we pioneer a systematic solution. First, we propose **DAT**, the first open-world drone active air-to-ground tracking benchmark. It encompasses 24 city-scale scenes, featuring targets with human-like behaviors and high-fidelity dynamics simulation. DAT also provides a digital twin tool for unlimited scene generation. Additionally, we propose a novel reinforcement learning method called **GC-VAT**, which aims to improve the performance of drone tracking targets in complex scenarios. Specifically, we design a Goal-Centered Reward to provide precise feedback across viewpoints to the agent, enabling it to expand perception and movement range through unrestricted perspectives. Inspired by curriculum learning, we introduce a Curriculum-Based Training strategy that progressively enhances the tracking performance in complex environments. Besides, experiments on simulator and real-world images demonstrate the superior performance of GC-VAT, achieving a Tracking Success Rate of approximately 72% on the simulator. The benchmark and code are available at https://github.com/SHWplus/DAT_Benchmark.

1 Introduction

Visual Active Tracking (VAT) aims to autonomously follow a target object by controlling the motion system of the tracker based on visual observations [80, 75]. It is widely used in real-world applications such as drone target tracking and security surveillance [22, 73, 77, 54]. Unlike passive visual tracking [3, 74, 33, 5, 9, 84, 67, 58], which involves proposing a 2D bounding box for the target on a frame-by-frame with a fixed camera pose, VAT actively adjusts the camera position to maintain the target within the field of view. Passive visual tracking often falls short in real-world scenarios due to the highly dynamic nature of most targets. Thus, VAT offers a more practical yet challenging solution for effective tracking in dynamic environments.

Recently, VAT methods have evolved into two main categories: pipeline VAT methods [40, 46, 15] and reinforcement learning-based VAT methods [19, 39, 18, 80]. **Pipeline VAT methods** employ a sequential framework where the visual tracking [32, 4, 62, 33] and control models are connected in series. The tracking model estimates the target position in the input image and the control model

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

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

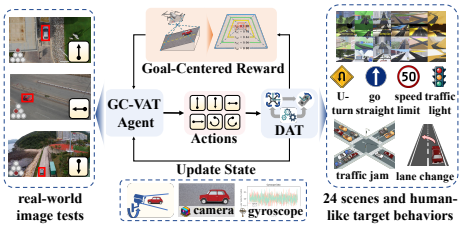


Figure 1: A pipeline for drone VAT.

Table 1: Comparison of DAT benchmark with simulators where existing methods are located.

	AD-VAT+ [80]	D-VAT [19]	AOT [39]	**DAT**
Scenes	8	4	2	**24**
Targets	1	1	1	**24**
Tracker	Ground	Drone	Ground	**Both**
Dynamics	✗	Simplified	✗	**Full Physics**
Target Behavior	Policy-based	Rule-based	Rule-based	**Human-like**
Scene Building	Manual	Manual	Manual	**Digital Twin**

generates control signals. While this modular design allows for clear task separation, it often requires significant manual effort to label the training data, and the control module requires additional tuning for different scenes. To address these issues, **reinforcement learning-based VAT methods** integrate visual tracking and control within a unified framework. These methods eliminate the need for separate tuning of the tracking and control modules by using a unified framework to map raw visual inputs directly to control actions. Therefore, the reinforcement learning-based VAT methods simplify system design and increase the efficiency of learning adaptive tracking behaviors in dynamic environments.

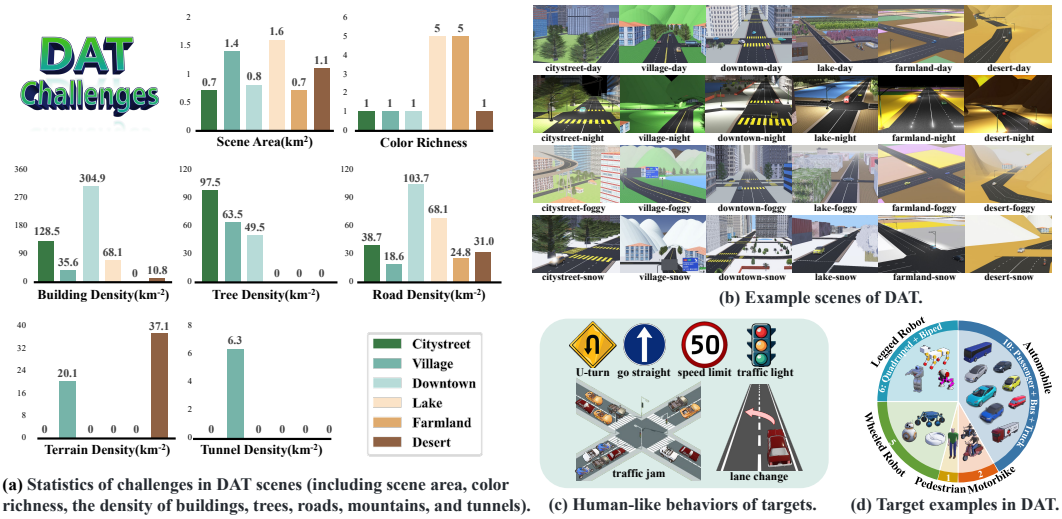
Unfortunately, achieving accurate drone VAT with reinforcement learning remains challenging, partly for the following reasons. **1) Missing unified benchmark.** Existing benchmark scenes are low in complexity, neglect tracker dynamics or rely on overly simplified models, making them inadequate to validate the agent performance (see Table 1). Previous methods [39, 19, 57] use rule-based target management, far from producing human-like target behaviors. Additionally, current 3D scenes are all manually constructed, leading to a heavy workload and limited scene number. **2) Vast environments with complex interference.** Open-world tracking involves large, dynamic environments with frequent interference. In previous methods [39, 19], trackers can only capture images from a fixed horizontal viewpoint. However, the fixed forward viewpoint captures excessive sky, reducing target-related visual information, especially for air-to-ground tracking tasks. Besides, since VAT goal is to keep the target at the image center, such viewpoint restricts the tracker to the same height as the target, severely limiting the perception and movement range. Moreover, training directly in complex conditions leads to slow convergence or difficulty in building strong behaviors.

To address the above limitations, we **first** propose **DAT**, the first open-world active air-to-ground tracking benchmark that simulates real-world complexity (see Fig. 2(b)). Specifically, DAT provides 24 city-scale scenes, full-fidelity simulations of drone dynamics, and a lightweight tool that can be integrated into any 3D scene to enable human-like target behaviors. It also offers a digital twin tool that can generate unlimited 3D scenes from real-world environments, enabling unlimited scene expansion. **Second**, we propose a novel drone VAT with reinforcement learning method (called **GC-VAT**), aiming to improve adaptability in complex and diverse scenarios. Specifically, we design a Goal-Centered Reward to provide precise feedback across viewpoints, enabling the agent to expand perception range through unrestricted perspectives. Besides, we propose qualitative and theoretical methods to analyze the effectiveness of our reward. In addition, inspired by curriculum learning [65, 41, 83], we propose a Curriculum-Based Training strategy that progressively improves agent performance in complex environments. Our contributions are summarized as follows:

1) **A comprehensive drone active tracking benchmark.** We present DAT benchmark, featuring high-fidelity dynamics, 24 city-scale scenes, and tools for simulating human-like target behaviors and unlimited scenes generation, enabling rigorous algorithm validation. 2) **A novel drone active tracking method.** We propose GC-VAT, which leverages a Goal-Centered Reward function and a Curriculum-Based Training strategy to enhance drone tracking performance in complex and dynamic environments. Besides, we propose qualitative and theoretical methods to analyze the effectiveness of our reward. 3) **Extensive experimental validation.** Experiments on simulator and real-world images validate DAT usability and GC-VAT effectiveness, with GC-VAT achieving a Tracking Success Rate of approximately 72% on the simulator.

2 Task Definition of Drone Active Tracking

DAT task seeks to train a model to control a drone for active target tracking in dynamic environments (see Fig. 1). Using visual and motion sensor data, the model learns actions to keep the target centered in view, ensuring robust performance across diverse scenarios.



(a) Statistics of challenges in DAT scenes (including scene area, color richness, the density of buildings, trees, roads, mountains, and tunnels).

(b) Example scenes of DAT.

(c) Human-like behaviors of targets.

(d) Target examples in DAT.

Figure 2: Statistics and simulator component examples of DAT. (a) Statistics on 7 complexity aspects in DAT scenes. (b) Example scenes of DAT. (c) Diverse behaviors of targets. (d) Examples of the tracking targets. More details can be found at https://github.com/SHWplus/DAT_Benchmark.

Observation spaces. The target is initially positioned at the center of the field of view, and the observation space comprises data acquired from sensors (e.g., RGB images with 84×84 resolution).

Action spaces. The action space can be either discrete or continuous. A discrete space defines a set of predefined drone maneuvers, whereas a continuous space allows direct control over the velocity.

Success criterion of DAT task. We define a success criterion when the model can keep the target object, which is initially located at the center of view, in the middle of the image for a long duration.

Challenges. Open-world drone active tracking is challenging due to limited data and high risks of trial-and-error in the real world, necessitating complex simulation environments. Additionally, the complexity and dynamics of open-world scenes further demand robust agent performance.

3 DAT Benchmark with Diverse Settings

We develop DAT, including 24 city-scale scenes built by an unlimited scene generation tool, high-fidelity drone dynamics simulation, and a versatile pipeline for producing human-like target behaviors.

3.1 Diverse Scene Construction

Digital twin tool. Users can select any region from *OpenStreetMap* [1] to obtain countless scenes using our tool. Specifically, it generates a high-precision road network with traffic lights and rules, and it converts elevation and vegetation data into 3D assets placed in the scene. Moreover, all assets in the generated scene are editable, allowing for data augmentation. See Appendix B for details.

Scene construction. Based on the above tool, we construct 6 outdoor scenes under 4 weather conditions, modeling 7 real-world complexities. Specifically, the *scene area*, *building density*, and *color richness* depict the complexity of the visual background. *Road density* and *terrain density* affect the target behaviors. The *tree density* and *tunnel density* measure the level of visual occlusion. As shown in Fig. 2(a), six scenarios exhibit unique and realistic complexity across the seven aspects:

- **Citystreet scene** covers an area of 0.7 square kilometers. It has a road density of 38.7 and a tree density of 97.5, mainly testing the agent's efficiency against tree occlusions.
- **Village scene** spans 1.4 square kilometers. This scene features a mountain density of 20.1 and a tunnel density of 6.3, requiring the agent to predict the target's movement when it is fully obscured.
- **Downtown scene** covers 0.8 square kilometers. It includes complex road elements and high building density of 304.9, challenging the agent's tracking accuracy and obstacle avoidance abilities.

• **Lake scene** encompasses 1.6 square kilometers. The density of road elements is 68.1, and the richness of background colors is 5, challenging the robustness across varying features and colors.

• **Farmland scene** covers an area of 0.7 square kilometers. The color richness is 5 and multiple color patches, challenging the agent's adaptability to multi-color environments.

• **Desert scene** covers 1.1 square kilometers. It includes a mountain density of 37.1 and a road density of 31.0. Some roads are covered by sand, testing the agent's adaptability to such conditions.

Four weather conditions are designed to test the agent's cross-domain adaptability. **Foggy** reduces visibility, **night** reduces brightness, and **snow** alters the color. The above 24 scenes (see Fig. 2(b)) can fully measure the agent tracking performance. See Appendix B for scene construction details.

3.2 Various Trackers and Targets Construction

Drone Active Tracking in the real world involves diverse targets depending on tasks. DAT provides diverse targets with human-like behaviors and enables high-fidelity tracker dynamics simulation.

Tracker. DAT benchmark supports two tracker types: drones and ground robots. The drone used is the *DJI Matrice 100* [14], equipped with a *3-axis gimbal*, allowing for precise camera adjustments. Unlike simpler kinematic models in [19] and methods that ignore the dynamics[39], DAT leverages *webots* [37] to simulate the drone's full dynamics, including mass, inertia, aerodynamics, and the response and jitter of the gimbal, closely matching real drones. See Appendix B for details.

Targets. DAT includes five categories of targets: *automobile*, *motorbike*, *pedestrian*, *wheel robot*, and *legged robot*, with a total of 24 tracking targets (see Fig. 2(d)). See Appendix B for details.

Target Management. We propose a novel pipeline to simulate realistic target behavior. Specifically, DAT first utilizes road networks generated by the tools described in Section 3.1, and directly integrates them with the SUMO traffic simulator [36]. Then, random trajectories are assigned to each vehicle, with SUMO managing its motion. To bridge the gap between simulation and visualization, we implement a controller that translates motion data into human-like driving behaviors for 3D vehicles (see Fig. 2(c)). Our controller also adheres to traffic rules and can simulate phenomena such as traffic light waits and traffic jams. Even better, the controller can be applied to any 3D scene.

4 VAT with Reinforcement Learning

In this paper, we primarily focus on visual active tracking (VAT), a core task within DAT benchmark. We propose a drone visual active tracking with reinforcement learning method called Goal-Centered-VAT (**GC-VAT**), aiming to improve the performance of tracking targets in complex scenes. As shown in Fig. 1, we model drone active tracking as a Markov Decision Process (MDP) and train a Drone Agent capable of adapting to unrestricted viewpoint conditions to track a target in the open scene.

4.1 MDP for Drone Active Tracking

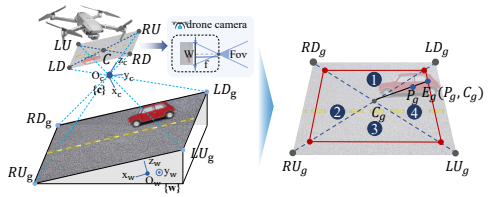
We seek to learn end-to-end drone tracking policies in dynamic environments by modeling the task as an MDP: $\langle \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{R}}, \boldsymbol{\gamma}, \boldsymbol{\mathcal{T}} \rangle$. In this representation, $\boldsymbol{\mathcal{S}}$ denotes the state space, $\boldsymbol{\mathcal{A}}$ represents the action space, and $\boldsymbol{\gamma}$ is the discount factor. At each time step t, the agent takes the state $s_t \in \boldsymbol{\mathcal{S}}$ as input and performs an action $a_t \in \boldsymbol{\mathcal{A}}$. Next, the simulator transitions to the next state $s_{t+1} = \mathcal{T}(s_t, a_t)$ and calculates the reward $r_t = \mathcal{R}(s_t, a_t)$ for the current step. The details of the MDP are as follows:

State $\boldsymbol{\mathcal{S}}$ is the visual information of the scene. At each time step t, the camera captures one image of size 84×84 as the current state.

Action $\boldsymbol{\mathcal{A}}$ is a set of discrete actions, including *forward*, *backward*, *leftward*, *rightward*, *turn left*, *turn right*, and *stop* movements. At each time step, the Drone Agent selects an action $a_t \in \boldsymbol{\mathcal{A}}$ based on the state s_t and actively controls the camera movement.

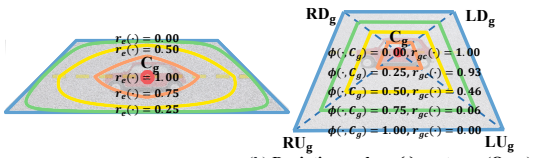
Transition $\mathcal{T}(s_t, a_t)$ is a function $\boldsymbol{\mathcal{T}} : \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \to \boldsymbol{\mathcal{S}}$ that maps s_t to s_{t+1}. In this paper, we use the *webots* dynamics engine to provide a realistic transition function.

Reward $\mathcal{R}(s_t, a_t)$ is the reward function. The goal-centered rewards are given in Section 4.2.



(a) Camera projection and pinhole imaging (b) Reward acquisition

Figure 3: Diagram of reward acquisition.



(a) Reward contours of D-VAT (b) Deviation and $r_{gc}(\cdot)$ contours (Ours)

Figure 4: Reward design analysis diagram.

Network structure of Drone Agent. Similar to previous works [39, 80], we select a backbone architecture consisting of a CNN followed by a GRU network [11] (see Appendix C.2).

Key Challenges in Drone Active Tracking. In open-world environments, drones face unpredictable target behaviors and frequent occlusions. Designing a single reward that encourages diverse and robust tracking actions is extremely difficult. To address this, we propose a **goal-centered reward** in Section 4.2. Moreover, given the vast observation space, discovering successful policies is non-trivial. To facilitate efficient learning, we introduce a **curriculum-based training strategy** in Section 4.4.

4.2 Goal-Centered Reward Design

For drone tracking a ground-based target, our reward is designed to characterize the target's position in the image and guide the drone to keep it centered. Therefore, we first need to select an appropriate distance metric to quantify the proximity between the target and the center in the image plane.

Since the drone typically captures images from a top-down perspective, the image plane is not parallel to the ground. Due to the affine transformation, the projection of the image plane becomes a trapezoid (see Fig. 3(b)), and the physical distance between the drone and the target cannot directly correspond to their pixel distance. Existing methods [39, 19] compute the Euclidean distance between the drone and the target, which may not accurately reflect their spatial relationship in the image plane.

To address this issue, we employ a deviation metric $\phi(\cdot,\cdot)$ to measure the distance between the target and the image center projection, as illustrated in Fig. 3(b). Specifically, given a target point P_g and the image center projection C_g, the deviation metric is computed by

$$\phi(P_g, C_g) = \frac{\mid P_g - C_g \mid}{\mid E_g(P_g, C_g) - C_g \mid}, \tag{1}$$

where $|P_g - C_g|$ denotes the distance from P_g to C_g and $|E_g(P_g, C_g) - C_g|$ represents distance from point $E_g(P_g, C_g)$ to the center. $E_g(P_g, C_g)$ is the intersection of the line connecting P_g and C_g with the projected image boundary, as shown in Fig. 3(b).

The deviation $\phi(\cdot,\cdot)$ is designed to ensure that targets inside the image are closer to the center than those outside, with contours shown in Fig. 4(b).

Principles for Reward Design. The objective of the VAT task is to keep the target at the image center. Thus, the targets closer to the image center projection should get higher reward values. For deviation metric $\phi(\cdot,\cdot)$, the design principle of the reward function $\mathcal{R}_\phi(\cdot)$ is defined as:

$$\forall P_1, P_2 \in \mathcal{W}, \text{ if } \phi_1 < \phi_2, \text{ then } \mathcal{R}_\phi(\phi_1) > \mathcal{R}_\phi(\phi_2), \tag{2}$$

where $\mathcal{W}$ denotes the valid region with non-zero reward, and ϕ_1, ϕ_2 represents the deviation from the target point to the image center projection.

Goal-Centered Reward Function. Our reward $r_{gc}(\cdot)$ decreases as the target moves away from the projected image center C_g, and is zero if outside, as shown as follows:

$$r_{gc}(P_g) = \begin{cases} \tanh\left(\alpha(1-\phi(P_g, C_g))^3\right), & P_g \in \mathcal{I}_{clip} \\ 0, \quad otherwise \end{cases}. \tag{3}$$

The attenuation degree of $r_{gc}(\cdot)$ can be adjusted using the hyperparameter α, set to 4. The $\tanh(\cdot)$ provides a strong indication of the task goal due to its relatively quick decay at the image center. $\mathcal{I}_{clip}$ is the truncated image range set to prevent the drone from keeping the target at the edge of the image. The truncation of the image can be controlled using the hyperparameter λ_{clip} as: $\lambda_{clip} = H_{\mathcal{I}_{clip}}/H$, where H and $H_{\mathcal{I}_{clip}}$ are the heights of the original and the truncated image. We set $\lambda_{clip} = 0.7$.

Algorithm 1 Curriculum-Based Training (CBT)

Input: Initial policy parameters θ_0, phase threshold η, total steps N, rollout steps n
Initialize: Training phase $phase \leftarrow 1$, reward buffer $\mathcal{B} \leftarrow \emptyset$, rollout buffer $\mathcal{B}_r \leftarrow \emptyset$
for each step $k = 0, 1, ..., N-1$ **do**
  if $phase = 1$ **then**
    Configure simple environment: linear target trajectories + no obstacles
  else
    Configure complex environment: varied target movements + obstacles/occlusions
  Collect transition $\tau_k = (s_t, s_{t+1}, a_t, r_t)$ with rewards calculated via (3)
  Append to buffer: $\mathcal{B} \leftarrow \mathcal{B} \cdot r_k$, $\mathcal{B}_r \leftarrow \mathcal{B}_r \cdot \tau_k$
  if $k \mod n = 0$ **then**
    Update policy using PPO: $\theta_{k+1} \leftarrow$ PPO_Update$(\theta_k, \mathcal{B}_r)$
    Clear rollout buffer: $\mathcal{B}_r \leftarrow \emptyset$
  if $phase = 1$ **and** $\frac{1}{|\mathcal{B}|}\sum_{r_t \in \mathcal{B}} r_t \geq \eta$ **then**
    Switch training phase: $phase \leftarrow 2$
    Clear buffer: $\mathcal{B} \leftarrow \emptyset$, $\mathcal{B}_r \leftarrow \emptyset$

More details about the Goal-Centered Reward. The reward function (Eq. 3) relies on the projections of the four corners and image center to compute deviation $\phi(\cdot, \cdot)$. As shown in Fig. 3(a), in the camera frame $\{\mathrm{c}\}$, the image center and four corner points have the coordinates $C(-f, 0, 0)$, $LU(-f, -\frac{1}{2}W, \frac{1}{2}H)$, $LD(-f, -\frac{1}{2}W, -\frac{1}{2}H)$, $RU(-f, \frac{1}{2}W, \frac{1}{2}H)$, $RD(-f, \frac{1}{2}W, -\frac{1}{2}H)$, where W and H are the image width and height and f denotes the camera focal length, which can be computed using the pinhole imaging principle [8] as: $f = \frac{W}{2\tan(\frac{1}{2}FoV)}$. FoV is the camera field of view. Next, the equations of the lines connecting the image center and the four corner points to the optical center $O_c(0, 0, 0)$ can be obtained in frame $\{c\}$(light blue dashed lines in Fig. 3(a)):

$$\begin{cases} l_{LUO_c} : \frac{x}{-f} = \frac{2y}{-W} = \frac{2z}{H} \\ l_{LDO_c} : \frac{x}{-f} = \frac{2y}{-W} = \frac{2z}{-H} \\ l_{RUO_c} : \frac{x}{-f} = \frac{2y}{W} = \frac{2z}{H} \\ l_{RDO_c} : \frac{x}{-f} = \frac{2y}{W} = \frac{2z}{-H} \\ l_{CO_c} : y = 0, z = 0 \end{cases}, \tag{4}$$

where l_{LUO_c} is the line connecting LU to O_c, similarly for l_{LDO_c}, l_{RUO_c}, l_{RDO_c} and l_{CO_c}. Thus, the projections of the points can be obtained by intersecting the lines with the ground plane.

Therefore, we next derive the expressions for the ground plane and the target. For clarity, we adopt a unified representation in frame $\{c\}$. In DAT scenes, the road surfaces are smooth. Thus, in the world frame $\{w\}$ (see Fig. 3(a)), the ground plane G_w is defined as: $z = h$, where h denotes the ground height. For simplicity, we here express G_w in $\{c\}$ as G_c: $A_g x + B_g y + C_g z + D_g = 0$, with A_g, B_g, C_g, D_g derived in Appendix C.2. Furthermore, the target coordinates $P_v = (x_v, y_v, z_v, 1)^T$ in $\{w\}$ can be transformed to $\{c\}$ using *homogeneous transformation matrix* [7] T_{cw}: $P_g = T_{cw}^{-1} P_v$.

Subsequently, the ground projections can be obtained by intersecting lines in Eq. 4 and G_c:

$$\begin{cases} LU_g : (-f, -\frac{1}{2}W, \frac{1}{2}H)t_{lu}, & t_{lu} = D_g(A_g f + \frac{1}{2}B_g W - \frac{1}{2}C_g H)^{-1} \\ LD_g : (-f, -\frac{1}{2}W, -\frac{1}{2}H)t_{ld}, & t_{ld} = D_g(A_g f + \frac{1}{2}B_g W + \frac{1}{2}C_g H)^{-1} \\ RU_g : (-f, \frac{1}{2}W, \frac{1}{2}H)t_{ru}, & t_{ru} = D_g(A_g f - \frac{1}{2}B_g W - \frac{1}{2}C_g H)^{-1} \\ RD_g : (-f, \frac{1}{2}W, -\frac{1}{2}H)t_{rd}, & t_{rd} = D_g(A_g f - \frac{1}{2}B_g W + \frac{1}{2}C_g H)^{-1} \\ C_g : (-\frac{D_g}{A_g}, 0, 0) \end{cases}, \tag{5}$$

where LU_g, LD_g, RU_g, RD_g and C_g are the projections of LU, LD, RU, RD and C. Using target coordinates P_g and Eq. 5, the reward is computed as Eq. 3. See Appendix C.2 for details.

4.3 Theoretical Guarantees on Reward Design

Existing methods [39, 19] assume a fixed forward camera view and use distance-based rewards. However, when the view changes, these rewards may fail due to the affine transformation effect in

image projection. We hereby provide a theoretical analysis to show that commonly used distance-based rewards will fail when the camera deviates from a fixed horizontal forward view.

To this end, we define $\mathcal{R}_d(\cdot)$ as a distance-based reward using Euclidean distance between the target and the image center projection. A distance-based reward $\mathcal{R}_d(\cdot)$ satisfying Eq. 2 may still assign higher rewards to targets farther from the center under the metric $\phi(\cdot,\cdot)$, rendering it ineffective. In contrast, any deviation-based reward $R_\phi(\cdot)$ satisfying Eq. 2 can effectively reflect the target position.

Proposition 1 *The commonly used Euclidean distance $d(\cdot,\cdot)$ between the target and the image center proposition does not align with the deviation $\phi(\cdot,\cdot)$ of the target from the image center projection, when the camera is not at a fixed horizontal forward viewpoint. That is:*

$$\exists P_1, P_2 \in \mathcal{I}_p, \textit{ s.t. } \phi_1 < \phi_2, \; d(P_1, C_g) > d(P_2, C_g), \tag{6}$$

where $\phi_i = \phi(P_i, C_g)$, P_i are points in the projection region $\mathcal{I}_p$, C_g is the image center projection. See Appendix C.1 for theoretical proof.

Remark 1. A distance-based reward $\mathcal{R}_d(\cdot)$ satisfying Eq. 2 results in targets closer to the center receiving lower rewards, when the camera is not at a fixed horizontal forward viewpoint. That is:

$$\exists P_1, P_2 \in \mathcal{I}_p, \; s.t. \; \phi_1 < \phi_2, \; \mathcal{R}_d(d_1) < \mathcal{R}_d(d_2), \tag{7}$$

where $d_i = d(P_i, C_g)$, and $\phi_i = \phi(P_i, C_g)$. This illustrates the failure of the distance-based reward under these viewpoints. See Appendix C.1 for theoretical proof.

Qualitative Analysis. According to the **Theoretical Analysis** above, rewards should decrease monotonically along the deviation contours in Fig. 4(b) as the target moves toward the projection boundary. Thus, the reward contours must align with the deviation contours. The contours of $r_{gc}(\cdot)$ in Fig. 4(b) perfectly align, indicating accurate position feedback. In contrast, D-VAT [19] (see Fig. 4(a)) shows misaligned contours, explaining its failure as noted in **Remark 1**.

4.4 Training with Curriculum Learning

DAT scenes contain numerous dynamic targets and obstacles, hindering convergence and performance. Progressively training the agent from simpler to more complex environments enhances performance and accelerates learning for the final task [63]. Therefore, we propose a Curriculum-Based Training (CBT) strategy to optimize reinforcement learning training in complex environments.

To address the challenges, we employ the Proximal Policy Optimization (PPO) [55] algorithm, known for its efficiency in control tasks. To further enhance agent adaptability and robustness, we apply domain randomization during agent training. Specifically, we randomize the drone's initial position and orientation relative to the target to promote diverse behaviors. Additionally, we randomize the gimbal pitch angle to improve the agent's spatial perception. See Appendix C.2 for further details.

Given the scene complexity, we adopt a CBT strategy, which divides the model training into two stages. The first stage consists of a simplified environment with straight line target trajectories and no obstacles. The agent learns to center the target through the reward r_t in Eq. 3. In the second stage, the agent encounters more varied target movements and complex visual information, such as tree occlusions and crosswalks. The goal of the agent is to develop stronger generalization abilities based on task understanding in the first stage. See Algorithm 1 for the pseudocode of the CBT strategy.

5 Experiments

5.1 Experimental Settings

Experimental Setup. We conduct cross-scene and cross-domain tests. The former tests an agent trained under daytime conditions in unseen scenes with the same weather. The latter evaluates the agent in the same scene under varying weather conditions. See Appendix E.1 for details.

Metrics. We use cumulative reward ($CR = \sum_{t=1}^{E_l} r_{gc}$) and tracking success rate ($TSR = \frac{1}{E_{ml}} \sum_{t=1}^{E_l} r_{dt} \times 100\%$) to evaluate the agent performance. CR primarily reflects how well the agent centers the target over episode length E_l, while TSR measures the ability to keep the target in view, with $r_{dt} = 1$ meaning the target is within the view (See Appendix C), and E_{ml} denoting the

Table 2: Results of within and across scenes on DAT benchmark.

Method	*citystreet*		*desert*		*village*		*downtown*		*lake*		*farmland*	
	CR	TSR	CR	TSR	CR	TSR	CR	TSR	CR	TSR	CR	TSR
Within Scene												
AOT	$49_{\pm 3}$	$0.25_{\pm 0.02}$	$9_{\pm 1}$	$0.06_{\pm 0.00}$	$46_{\pm 5}$	$0.23_{\pm 0.03}$	$54_{\pm 5}$	$0.29_{\pm 0.01}$	$47_{\pm 3}$	$0.24_{\pm 0.02}$	$60_{\pm 25}$	$0.23_{\pm 0.01}$
D-VAT	$48_{\pm 8}$	$0.26_{\pm 0.02}$	$47_{\pm 13}$	$0.26_{\pm 0.04}$	$44_{\pm 8}$	$0.22_{\pm 0.05}$	$9_{\pm 1}$	$0.06_{\pm 0.01}$	$46_{\pm 8}$	$0.26_{\pm 0.06}$	$13_{\pm 1}$	$0.07_{\pm 0.00}$
Ours	$\mathbf{279}_{\pm 110}$	$\mathbf{0.80}_{\pm 0.30}$	$\mathbf{307}_{\pm 124}$	$\mathbf{0.84}_{\pm 0.29}$	$\mathbf{239}_{\pm 134}$	$\mathbf{0.73}_{\pm 0.32}$	$\mathbf{203}_{\pm 119}$	$\mathbf{0.65}_{\pm 0.30}$	$\mathbf{181}_{\pm 116}$	$\mathbf{0.61}_{\pm 0.31}$	$\mathbf{243}_{\pm 117}$	$\mathbf{0.68}_{\pm 0.32}$
Cross Scene												
AOT	$48_{\pm 5}$	$0.24_{\pm 0.02}$	$9_{\pm 0}$	$0.06_{\pm 0.00}$	$52_{\pm 11}$	$0.25_{\pm 0.03}$	$52_{\pm 6}$	$0.28_{\pm 0.03}$	$48_{\pm 5}$	$0.24_{\pm 0.02}$	$49_{\pm 7}$	$0.24_{\pm 0.03}$
D-VAT	$49_{\pm 9}$	$0.26_{\pm 0.04}$	$48_{\pm 8}$	$0.27_{\pm 0.03}$	$50_{\pm 14}$	$0.25_{\pm 0.06}$	$8_{\pm 1}$	$0.05_{\pm 0.00}$	$51_{\pm 14}$	$0.25_{\pm 0.06}$	$14_{\pm 1}$	$0.07_{\pm 0.01}$
Ours	$\mathbf{144}_{\pm 111}$	$\mathbf{0.52}_{\pm 0.29}$	$\mathbf{229}_{\pm 115}$	$\mathbf{0.67}_{\pm 0.27}$	$\mathbf{156}_{\pm 119}$	$\mathbf{0.55}_{\pm 0.31}$	$\mathbf{201}_{\pm 121}$	$\mathbf{0.64}_{\pm 0.30}$	$\mathbf{163}_{\pm 115}$	$\mathbf{0.51}_{\pm 0.29}$	$\mathbf{162}_{\pm 106}$	$\mathbf{0.54}_{\pm 0.26}$

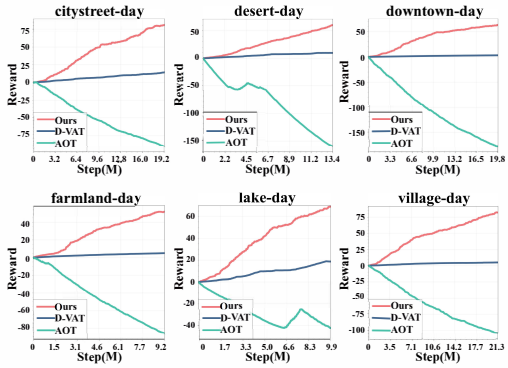


Figure 5: Reward values during training.

Table 3: Results of cross domain on DAT.

Method	*night*		*foggy*		*snow*	
	CR	TSR	CR	TSR	CR	TSR
AOT	$42_{\pm 4}$	$0.22_{\pm 0.02}$	$44_{\pm 7}$	$0.22_{\pm 0.02}$	$44_{\pm 7}$	$0.22_{\pm 0.02}$
D-VAT	$35_{\pm 7}$	$0.19_{\pm 0.03}$	$37_{\pm 7}$	$0.19_{\pm 0.03}$	$34_{\pm 6}$	$0.20_{\pm 0.03}$
Ours	$\mathbf{217}_{\pm 125}$	$\mathbf{0.64}_{\pm 0.32}$	$\mathbf{243}_{\pm 114}$	$\mathbf{0.76}_{\pm 0.26}$	$\mathbf{178}_{\pm 105}$	$\mathbf{0.60}_{\pm 0.26}$

Table 4: Results of ablation experiments on DAT.

Method	*Within-Scene*		*Cross-Scene*		*Cross-Domain*	
	CR	TSR	CR	TSR	CR	TSR
$R_{\text{D-VAT}}$	$9_{\pm 1}$	$0.06_{\pm 0.00}$	$8_{\pm 1}$	$0.05_{\pm 0.00}$	$9_{\pm 0}$	$0.06_{\pm 0.00}$
w/o CBT	$46_{\pm 2}$	$0.23_{\pm 0.01}$	$53_{\pm 16}$	$0.26_{\pm 0.07}$	$46_{\pm 2}$	$0.23_{\pm 0.01}$
w/o AR	$106_{\pm 88}$	$0.44_{\pm 0.23}$	$92_{\pm 72}$	$0.37_{\pm 0.19}$	$80_{\pm 63}$	$0.36_{\pm 0.19}$
w/o HR	$174_{\pm 118}$	$0.49_{\pm 0.30}$	$148_{\pm 129}$	$0.48_{\pm 0.32}$	$184_{\pm 124}$	$0.57_{\pm 0.30}$
w/o VR	$211_{\pm 138}$	$0.63_{\pm 0.35}$	$161_{\pm 115}$	$0.54_{\pm 0.32}$	$203_{\pm 117}$	$0.60_{\pm 0.32}$
w/o PR	$139_{\pm 119}$	$0.61_{\pm 0.33}$	$124_{\pm 85}$	$0.48_{\pm 0.25}$	$145_{\pm 122}$	$0.52_{\pm 0.28}$
Ours	$\mathbf{243}_{\pm 117}$	$\mathbf{0.68}_{\pm 0.32}$	$\mathbf{162}_{\pm 106}$	$\mathbf{0.54}_{\pm 0.26}$	$\mathbf{222}_{\pm 110}$	$\mathbf{0.65}_{\pm 0.27}$

maximum episode length. Agents are initialized at four relative angles to the target ($[0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}]$ rad), with 10 episodes per angle (40 total). The mean and variance of these results are calculated for each map, and the final cross-scene and cross-domain performance are averaged across different scenes.

Baselines. We reproduce two SOTA methods: AOT [39] and D-VAT [19]. Both baselines and other methods [81, 18] use distance-based rewards. As concluded in Section 4.3, they may fail in tilted top-down views. Thus, these baselines sufficiently highlight GC-VAT superiority. See Appendix D.

5.2 Comparison Experiments

We compare our GC-VAT with the SOTA methods for within-scene performance and cross-scene cross-domain generalization performance on DAT benchmark. As shown in Fig. 5, our method achieves consistently higher and steadily increasing rewards throughout training, demonstrating its effectiveness. Both AOT and D-VAT methods fail to learn effective policies due to the misleading feedback from their distance-based rewards. In particular, AOT learns to quickly drive the target out of view, resulting in a rapidly declining reward curve. The results validate the theoretical analysis in Section 4.3. It is worth noting that although AOT and D-VAT exhibit low variance in their experimental results, consistently low rewards typically indicate a failure to learn effective tracking policies.

Within-scene performance. We train the model on all scenes and evaluate it on the original scene. Our GC-VAT performs significantly better than other methods as shown in Table 2. For the CR, the average performance improvement on six maps relative to the D-VAT method is $591\%(35 \rightarrow 242)$. Regarding the TSR, the average enhancement is $279\%(0.19 \rightarrow 0.72)$.

Cross-scene performance. Our method demonstrates strong cross-scene generalization, as shown in Table 2. Specifically, GC-VAT achieves a $376\%(37 \rightarrow 176)$ improvement in average CR and a $200\%(0.19 \rightarrow 0.57)$ improvement in average TSR compared to D-VAT.

Cross-domain performance. As shown in Table 3, our method outperforms existing methods significantly in cross-domain generalization. Specifically, GC-VAT demonstrates an average CR enhancement of $509\%(35 \rightarrow 213)$ relative to D-VAT and TSR boost of $253\%(0.19 \rightarrow 0.67)$.

Table 5: Performance under wind disturbances and target distractors.

	CR	*TSR*
w/ Forward	$302_{\pm 94}$	$0.91_{\pm 0.18}$
w/ Lateral	$304_{\pm 82}$	$0.91_{\pm 0.19}$
w/ Yaw	$301_{\pm 120}$	$0.88_{\pm 0.23}$
w/ Distractor	$293_{\pm 120}$	$0.91_{\pm 0.15}$
Ours	$\mathbf{316}_{\pm 84}$	$\mathbf{0.94}_{\pm 0.14}$

Table 6: Performance under rainy conditions and unseen targets. We evaluate the model trained on citystreet-day.

Method	*Within-Scene*		*Cross-Scene*		*Cross-Domain*	
	CR	*TSR*	*CR*	*TSR*	*CR*	*TSR*
w/ rain	$266_{\pm 110}$	$0.74_{\pm 0.29}$	$139_{\pm 109}$	$0.45_{\pm 0.30}$	$274_{\pm 103}$	$0.77_{\pm 0.29}$
Unseen Target	$222_{\pm 92}$	$0.79_{\pm 0.25}$	$131_{\pm 89}$	$0.50_{\pm 0.33}$	$207_{\pm 94}$	$0.79_{\pm 0.27}$
Ours	$\mathbf{279}_{\pm 110}$	$\mathbf{0.80}_{\pm 0.30}$	$\mathbf{144}_{\pm 111}$	$\mathbf{0.52}_{\pm 0.29}$	$\mathbf{258}_{\pm 110}$	$\mathbf{0.82}_{\pm 0.23}$

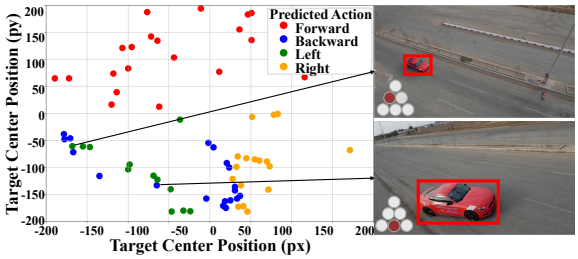


Figure 6: Results on real-world images.

Table 7: Effectiveness of GC-VAT on Sim2Real test. We select eight video sequences from each dataset for evaluation.

Video	*VOT* [30]	*DTB70* [35]	*UAVDT* [20]
Average Correct Action Rate			
Random	0.413	0.426	0.421
Ours	**0.795**	**0.833**	**0.802**

5.3 Ablation Experiments

We conduct ablation experiments on Goal-Centered Reward to validate the results of the analysis presented in Section 4.3. Moreover, we verify whether the Curriculum-Based Training strategy and domain rondomization from Section 4.4 lead to a significant performance improvement. We present results on the *farmland* map in Table 4, with additional results provided in Appendix E.3.

Effectiveness of reward design. We contrast the performance of GC-VAT method when using the reward defined in Eq. 3 and that in [19]. As shown in Table 4, significant performance enhancements (about 800% improvement in TSR across-scene and cross-domain) are evident with the utilization of Eq. 3. These results strongly corroborate the analysis in Section 4.3 and underscore the effectiveness of the proposed reward. See Appendix E.3 for more experimental results.

Effectiveness of CBT strategy and domain randomization. As shown in Table 4, without the CBT strategy, the model fails to learn effective tracking policies, resulting in consistently low rewards across different tests. In addition, our domain randomization approach yields significant improvements. Specifically, AR, HR, VR, and PR denote the randomization of the drone's initial angle, horizontal and vertical distance relative to the target, and gimbal pitch angle, respectively. Among these, AR contributes the most to performance gains, indicating that encouraging diverse actions through angle randomization facilitates the agent's exploration of optimal policies.

Robustness under wind gusts and precipitation. To further investigate the impact of real-world disturbances on the GC-VAT method, we conduct rigorous tests under wind gusts and sensor degradation caused by precipitation. Specifically, we simulate wind effects by applying randomized perturbations along the forward, lateral, and yaw directions during testing. The results are summarized in Table 5, where the model is trained on citystreet-day and evaluated on citystreet-foggy with added wind perturbations. The Tracking Success Rate (TSR) drops by less than 0.06, demonstrating that GC-VAT maintains strong robustness under significant wind disturbances. See Appendix E.3 for more details.

To simulate the blurring caused by raindrops, we follow established practices in test-time adaptation literature [34]. Specifically, we train the policy on citystreet-day map and evaluate under synthetically generated rain in within-scene, cross-scene, and cross-domain settings. To ensure realism, we exclude snowy conditions from the cross-domain evaluation, as snow and rain rarely co-occur in real-world environments. The results in Table 6 show only marginal performance degradation (less than 0.07 in TSR) under rain simulation, confirming that GC-VAT is robust to blurring caused by raindrops.

Robustness to distractors and novel targets. As shown in Table 5, our model maintains high tracking performance even when a similar-looking vehicle is introduced near the target, demonstrating its ability to effectively distinguish the true target from confusers. In addition, we evaluate GC-VAT on an unseen target class (bus). As shown in Table 6, our model maintains strong tracking performance, with a TSR drop of less than 0.03 when encountering this novel object.

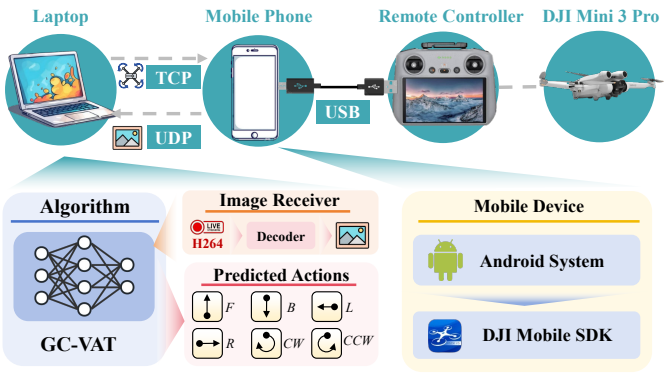


Figure 7: Schematic of the real-world deployment pipeline.

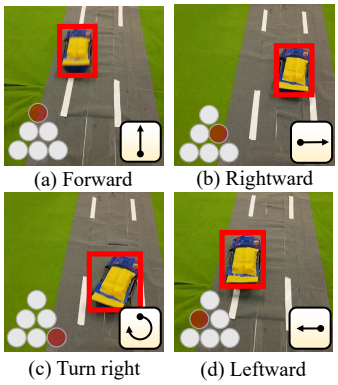


Figure 8: Results on real drones.

5.4 Experiments in Real-world Scenarios

Effectiveness on real-world images. Due to the difficulty real-robot evaluation, we follow [39] and validate GC-VAT on real images. We perform zero-shot transfer tests using 8 videos each from VOT [30], DTB70 [35] and UAVDT [20] datasets. Although camera control is unavailable in recorded videos, we can feed frames into the model and verify the reasonableness of its predicted actions.

The output actions for the VOT video *car16* are shown in Fig. 6. Each point represents the target position in the image, with colors indicating different actions. As Fig. 6 illustrates, when the target is located on the left (right) side, the tracker tends to move left (right), attempting to bring the target to the center. Quantitatively, we use the *Correct Action Rate*, i.e., the accuracy of predicted actions, to evaluate the performance. As shown in Table 7, GC-VAT achieves an average *Correct Action Rate (CAR)* of 81.0% across 24 videos, demonstrating its effectiveness. More importantly, it is significantly superior to random policy ($p < 0.001$) as verified by a t-test. See Appendix E.4 for more results.

Effectiveness on real drones. Furthermore, as a critical step beyond image-based evaluation, we conduct real-world experiments on a *DJI Mini 3 Pro* [13] drone. As shown in Fig. 7, we deploy GC-VAT on a laptop equipped with an RTX 3050 GPU and an Intel i5 CPU, use the DJI Mobile SDK [12] to obtain images, and control the drone with the predicted actions. The entire pipeline operates at over 30 FPS. As Fig. 8 illustrates, the model can output actions that maintain the target at the image center. Quantitatively, GC-VAT achieves an average zero-shot TSR of 88.4% and a CAR of 81.3%. This successful zero-shot Sim-to-Real transfer validates the practical applicability of our approach.

6 Conclusion and Potential Impacts

In this paper, we propose the first open-world drone active air-to-ground tracking benchmark, called DAT. DAT benchmark encompasses 24 city-scale scenes, featuring targets with human-like behaviors and high-fidelity dynamics simulation. DAT also provides a digital twin tool for unlimited scene generation. DAT benchmark has the potential to impact several key areas, including: 1) Forgetting in Reinforcement Learning, 2) Robustness in Reinforcement Learning, 3) Multi-Agent Reinforcement Learning, and 4) Sim-to-Real Deployment. Additionally, we propose a reinforcement learning-based drone tracking method called GC-VAT, aiming to improve the performance of drone tracking targets in complex scenarios. Specifically, we design a Goal-Centered Reward to provide precise feedback across viewpoints to the agent, enabling it to expand perception range through unrestricted perspectives. Then we propose qualitative and theoretical methods to analyze the reward effectiveness. Moreover, inspired by curriculum learning, we implement a Curriculum-Based Training strategy that progressively improves agent performance in increasingly complex scenarios. Experiments on the simulator and real-world images validate the analysis and demonstrate that our method is significantly superior to the SOTA methods.

Acknowledgements

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

References

[1] Openstreetmap. https://www.openstreetmap.org/. Accessed 2025.10.09.

[2] Coppelia Robotics AG. Coppeliasim (formerly v-rep). https://www.coppeliarobotics.com/, 2025. Software.

[3] 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.

[4] 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.

[5] 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.

[6] Xiuli Bi, Jinwu Hu, Bin Xiao, Weisheng Li, and Xinbo Gao. Iemask r-cnn: Information-enhanced mask r-cnn. *IEEE Transactions on Big Data*, 9(2):688–700, 2022.

[7] Sébastien Briot, Wisama Khalil, Sébastien Briot, and Wisama Khalil. Homogeneous transformation matrix. *Dynamics of Parallel Robots: From Rigid Bodies to Flexible Elements*, pages 19–32, 2015.

[8] Germain Chartier. *Introduction to optics*, volume 1. Springer, 2005.

[9] 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, June 2023.

[10] Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

[11] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[12] Da-Jiang Innovations. Dji mobile sdk, 2025. Accessed: 2025-10-09.

[13] Da-Jiang Innovations. Support for dji mini 3 pro. https://www.dji.com/cn/support/product/mini-3-pro, 2025. Accessed 2025.10.09.

[14] Da-Jiang Innovations. Support for matrice 100. https://www.dji.com/cn/support/product/matrice100, 2025. Accessed 2025.10.09.

[15] 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.

[16] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 436–454, Cham, 2020. Springer International Publishing.

[17] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Qinfeng Shi, Daniel Cremers, Ian D. Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taix'e. Mot20: A benchmark for multi object tracking in crowded scenes. *ArXiv*, abs/2003.09003, 2020.

[18] 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.

[19] 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*, 2024.

[20] 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)*, September 2018.

[21] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, Qinghua Hu, Jiayu Zheng, Tao Peng, Xinyao Wang, Yue Zhang, et al. Visdrone-sot2019: The vision meets drone single object tracking challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

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

[23] 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.

[24] Epic Games. Unreal engine 4. https://www.unrealengine.com/, 2025. Software.

[25] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[26] Jinwu Hu, Yufeng Wang, Shuhai Zhang, Kai Zhou, Guohao Chen, Yu Hu, Bin Xiao, and Mingkui Tan. Efficient dynamic ensembling for multiple LLM experts. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 8095–8103. ijcai.org, 2025.

[27] Wei-Ming Hu, Qiang Wang, Jin Gao, Bing Li, and Stephen Maybank. Dcfnet: Discriminant correlation filters network for visual tracking. *Journal of Computer Science and Technology*, 39(3):691–714, 2024.

[28] Bo Huang, Jianan Li, Junjie Chen, Gang Wang, Jian Zhao, and Tingfa Xu. Anti-uav410: A thermal infrared benchmark and customized scheme for tracking drones in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[29] PTC Inc. Creo. https://www.ptc.com/en/products/creo. Accessed 2025.10.09.

[30] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, 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, Nov 2016.

[31] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[32] 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.

[33] 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.

[34] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1633–1642, 2019.

[35] Siyi Li and Dit-Yan Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[36] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[37] Cyberbotics Ltd. Webots: Open source mobile robot simulation software. https://cyberbotics.com/, 2025. Software.

[38] Sha Luo, Hamidreza Kasaei, and Lambert Schomaker. Accelerating reinforcement learning for reaching using continuous curriculum learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[39] 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.

[40] 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.

[41] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[42] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1928–1937. PMLR, 2016.

[43] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 445–461. Springer, 2016.

[44] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1387–1395, 2017.

[45] Keung Or, Kehua Wu, Kazashi Nakano, Masahiro Ikeda, Mitsuhito Ando, Yasuo Kuniyoshi, and Ryuma Niiyama. Curriculum-reinforcement learning on simulation platform of tendon-driven high-degree of freedom underactuated manipulator. *Frontiers in Robotics and AI*, 10:1066518, 2023.

[46] 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.

[47] Luis Patino, Tom Cane, Alain Vallee, and James Ferryman. Pets 2016: Dataset and challenge. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1240–1247, 2016.

[48] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 145–161. Springer, 2020.

[49] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[50] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[51] Liangliang Ren, Xin Yuan, Jiwen Lu, Ming Yang, and Jie Zhou. Deep reinforcement learning with iterative shift for visual tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–700, 2018.

[52] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[53] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[54] 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.

[55] John Schulman. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[56] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[57] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[58] 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.

[59] Ke Song, Wei Zhang, Ran Song, and Yibin Li. Online decision based visual tracking via reinforcement learning. In *Neural Information Processing Systems*, 2020.

[60] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

[61] Lars Væhrens, Daniel Díez Álvarez, Ulrich Berger, and Simon Bøgh. Learning task-independent joint control for robotic manipulators with reinforcement learning and curriculum learning. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1250–1257. IEEE, 2022.

[62] 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.

[63] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[64] Yufeng Wang, Jinwu Hu, Ziteng Huang, Kunyang Lin, Zitian Zhang, Peihao Chen, Yu Hu, Qianyue Wang, Zhuliang Yu, Bin Sun, Xiaofen Xing, Qingfang Zheng, and Mingkui Tan. Enhancing user-oriented proactivity in open-domain dialogues with critic guidance. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 8268–8276. ijcai.org, 2025.

[65] Yulin Wang, Yang Yue, Rui Lu, Yizeng Han, Shiji Song, and Gao Huang. Efficienttrain++: Generalized curriculum learning for efficient visual backbone training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[66] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *European conference on computer vision*, pages 107–122. Springer, 2020.

[67] 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.

[68] Jiayi Weng, Huayu Chen, Dong Yan, Kaichao You, Alexis Duburcq, Minghao Zhang, Yi Su, Hang Su, and Jun Zhu. Tianshou: A highly modularized deep reinforcement learning library. *Journal of Machine Learning Research*, 23(267):1–6, 2022.

[69] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[70] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[71] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[72] Weidai Xia, Dongming Zhou, Jinde Cao, Yanyu Liu, and Ruichao Hou. Cirnet: An improved rgbt tracking via cross-modality interaction and re-identification. *Neurocomputing*, 493:327–339, 2022.

[73] 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.

[74] 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.

[75] Di Yuan, Xiaojun Chang, Qiao Liu, Yi Yang, Dehua Wang, Minglei Shu, Zhenyu He, and Guangming Shi. Active learning for deep visual tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[76] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720, 2017.

[77] 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.

[78] Da Zhang, Hamid Maei, Xin Wang, and Yuan-Fang Wang. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*, 2017.

[79] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International journal of computer vision*, 129:3069–3087, 2021.

[80] 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.

[81] 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.

[82] Dong Zhou, Guanghui Sun, Wenxiao Lei, and Ligang Wu. Space noncooperative object active tracking with deep reinforcement learning. *IEEE Transactions on Aerospace and Electronic Systems*, 58(6):4902–4916, 2022.

[83] Pengfei Zhu, Jialu Li, Yu Wang, Bin Xiao, Jinglin Zhang, Wanyu Lin, and Qinghua Hu. Boosting pseudo-labeling with curriculum self-reflection for attributed graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024.

[84] 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.

Supplementary Materials for "Open-World Drone Active Tracking with Goal-Centered Rewards"

Contents

A Related Work

A.1 Passive Object Tracking

Most of the proposed visual tracking benchmarks belong to passive visual tracking. LaSOT [23] and OTB2015 [71] benchmarks contain a large number of ground-based videos. These benchmarks include target videos, and the tracking algorithms utilize both the video frames and the target labels for tracking. However, ground cameras tend to be affected by occlusion and suffer from the shortcoming of limited perceptual range, so the need for drone viewpoint tracking is gradually increasing in practical applications. UAV123 [43] and VisDrone2019 [21] benchmarks are proposed for drone viewpoint, expanding the spatial dimension of perception. Meanwhile, the single-object tracking benchmarks have difficulties for many targets. MOT20 [17] and TAO [16] benchmarks are proposed for multi-object tracking to solve the above problems. In addition, the above benchmarks include videos from the RGB camera. The RGB camera's recognition capabilities are limited in complex scenes, such as ocean environments, and challenging weather conditions, including nighttime and foggy. IPATCH [47] provides extra infrared images and other sensors like GPS to supplement the information of the sea scene. Huang et al. propose Anti-UAV410 [28], which provides infrared camera images for drone tracking.

Visual object tracking methods can be categorized into three main types: Tracking by Detection, Detection and Tracking (D&T), and pure tracking. Tracking by Detection methods [5, 69, 6] treat tracking as a sequence of independent detection tasks. These methods use object detection algorithms [50, 52] to identify the target object in each frame, connecting the detections through data association methods [31, 70] for continuous tracking. While effective in multi-target tracking, these methods

may suffer from high computational demands and issues with target occlusion. D&T approaches [66, 79, 48] integrate detection and tracking, creating end-to-end models that ensure seamless information flow and reduce redundant calculations through shared feature extraction networks. Pure tracking methods can be categorized into two main types: Correlation Filters (CF) [72, 44, 27] and Siamese Networks (SN) [32, 4, 62]. CF-based models train correlation filters on regions of interest, while SN-based models compare target templates with search areas to enable precise single-target tracking.

A.2 Visual Active Tracking

Passive visual tracking often falls short in real-world scenarios due to the highly dynamic nature of most targets. Visual Active Tracking (VAT) aims to autonomously follow a target object by controlling the motion system of the tracker based on visual observations [40, 80, 75]. Thus, VAT offers a more practical yet challenging solution for effective tracking in dynamic environments. Maalouf et al. [40] propose a two-stage tracking method (named FAn), which is based on a tracking model and a PID control model. This method accomplishes the fusion of perception and decision-making by transferring control information from the visual tracking model to the control model. However, the visual network necessitates extensive human labeling effort and the control model requires parameter adjustments for each scene, significantly constraining the model's generalizability. Recently, many approaches [39, 18, 80, 19] model the VAT task as a Markov Decision Process and employ end-to-end training with reinforcement learning, resulting in a significant enhancement of the agent's generalizability.

The complexity and diversity of VAT benchmarks are crucial for training agents with high generalizability. One common approach [19, 18, 80] to enhancing environmental diversity involves modifying texture features and lighting conditions within a single scene. However, these methods often result in low scene fidelity and unrealistic object placement. While UE4 [24] is used to create photorealistic environments in some benchmarks [80, 39], these benchmarks still face limitations in diversity and map size. Furthermore, the scenarios provided by these methods are often task-specific, offering limited configurability and lacking a unified benchmark for VAT tasks.

Existing approaches to VAT frequently neglect the randomness of target trajectories and the scalability of platforms. Target trajectories are typically predefined by rule-based patterns [19, 18, 39], which significantly restrict the exploration space. Zhong et al. [80] introduce learnable agents as targets, increasing trajectory randomness but adding additional cost. Most benchmarks provide only a single category of target [19, 18, 80, 39], limiting scalability and necessitating repetitive work for environment development. Zhou et al. [82] utilize CoppeliaSim [2] to provide five categories of noncooperative space objects. However, the use of a solid black background makes it unsuitable for general VAT scenarios. In contrast, our environment supports diverse, real-world target types and offers unified, lightweight management of target behaviors, ensuring both rationality and randomness in their actions.

A.3 Reinforcement learning in Visual Tracking

Reinforcement learning (RL) is widely used in large language models [26] and robot control [53] to improve exploration performance. It is also commonly applied in visual object tracking [76, 51, 78]. Song et al. [59] propose a decision-making mechanism based on hierarchical reinforcement learning (HRL), which achieves state-of-the-art performance while maintaining a balance between accuracy and computational efficiency. However, the actions generated by reinforcement learning in the aforementioned work cannot directly influence the camera's viewpoint, thereby failing to fully leverage the decision-making capabilities. Real-world applications increasingly require robust tracking in highly dynamic scenes, motivating researchers to explore reinforcement learning agents for effectively synchronizing visual perception and decision-making in VAT tasks. Dionigi et al. [19] demonstrate the feasibility of reinforcement learning for drone VAT missions. However, the assumption of a fixed-forward perspective limits its applicability in real-world tasks.

A.4 Curriculum Learning in Robot Control

Curriculum Learning (CL) is a training strategy that mimics a human curriculum by training models on simpler subsets of data at first and gradually expanding to larger and more difficult subsets of data

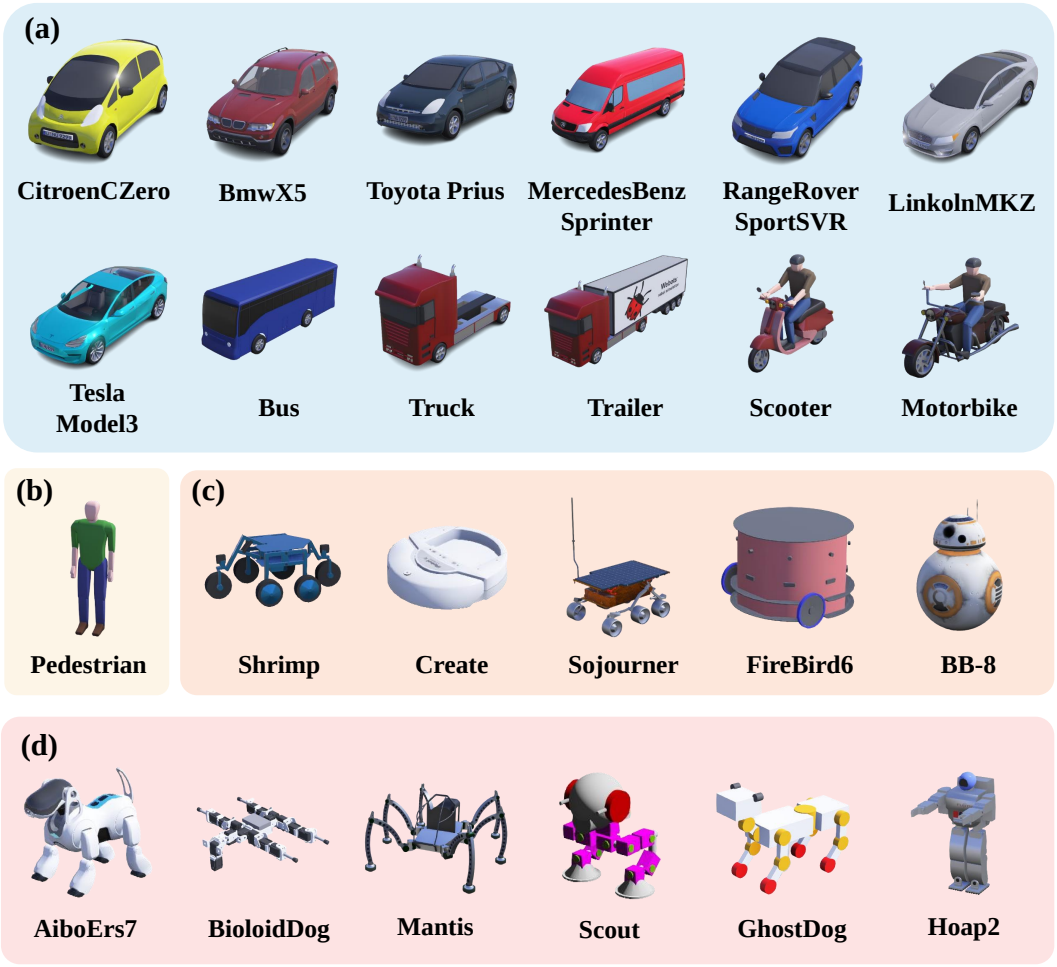


Figure 9: Examples of DAT benchmark targets. (a) Illustration of tracking targets for 10 types of *automobile* and 2 types of *motorbike*. (b) Illustration of tracking targets for the *pedestrian* type. (c) Illustration of tracking targets for 5 types of *wheeled robot*. (d) Illustration of tracking targets for 6 types of *legged robot*.

until they are trained on the entire dataset. CL is widely used in large language models [64] and robot control. As for robot control, reinforcement learning training is difficult due to the complexity of the training scenarios and the large action spaces. Therefore, curriculum learning is often required to reduce the difficulty of agent training. For instance, many works improve the walking ability of legged robots by adjusting terrain parameters through curriculum learning [53, 41]. Other studies improve the pushing and grasping performance of robotic arms by progressively increasing task difficulty [38, 61, 45].

In this paper, Curriculum Learning is introduced in the VAT task, and the training environment is transitioned from simple features to complex scenarios to achieve successful tracking of agent in complex outdoor environments.

B More Details of DAT Benchmark

More details of the digital twin tool. Our digital twin tool is based on the *osm_importer* tool in the *webots* simulation software. Users first need to download the map description file (*.osm* file) for a specific area from the *OpenStreetMap* website. Then, the tool preprocesses the map according to the configuration, modifying information such as the number of lanes and lane directions, and converts the processed file into a road network file (*.net.xml* file) that can be read by SUMO. Following this, the tool adds traffic lights and intersection traffic rules to the road network based on the configuration,

Table 8: State parameters of DAT benchmark.

Category	Sensor	Parameter	Type	Description	Potential Tasks
Vision	Camera	`Image`	Mat	Images	Default sensor
	LiDAR	`LidarCloud`	vector2000	Point cloud (m)	Obstacle avoidance
Motion	GPS	`Position`	vector3	Position (m)	Visual navigation
		`Linear`	vector3	Linear velocity (m/s)	Visual navigation
	Accelerometer	`Acc`	vector3	Acceleration (m/s^2)	Visual navigation
	Gyroscope	`Angular`	vector3	Angular velocity (rad/s)	Posture stabilization
	IMU	`Angle`	vector3	Euler angles (rad)	Posture stabilization
		`Orientation`	vector4	Quaternion representation	Posture stabilization

Table 9: Reward parameters of DAT benchmark. The homogeneous transformation matrices (HTM) $\mathtt{T}_{\mathtt{cw}}$ and $\mathtt{T}_{\mathtt{tw}}$ are 4×4 square matrices. Therefore, their data type double$[16]$ corresponds to a double array of length 16.

Parameter	Type	Description
`cameraWidth`	double	image width(px)
`cameraHeight`	double	image height(px)
`cameraFov`	double	camera field of view(rad)
`cameraF`	double	estimated camera focal length(px)
$\mathtt{T}_{\mathtt{cw}}$	double[16]	HTM of the camera relative to the world frame
$\mathtt{T}_{\mathtt{tw}}$	double[16]	HTM of the vehicle relative to the world frame
`cameraMidGlobalPos`	vector3d	ground projection of camera center mapped in the world frame
`carMidGlobalPos`	vector3d	coordinates of the vehicle center in the world frame
`cameraMidPos`	vector3d	coordinates of the camera center in the world frame
`carDronePosOri`	vector4d	1D orientation + 3D position of vehicle in the drone frame
`crash`	double	whether tracker collides with a building
`carDir`	double	car direction(*0-stop,1-go straight,2-turn left,3-turn right*)
`carTypename`	string	tracking target type

ensuring that the traffic flow operates correctly when the map is converted into a 3D scene. Finally, the tool reads the road, vegetation, and building information and converts them into *PROTO* assets for webots, which can then be correctly recognized and used by the DAT benchmark.

Scenario Construction. Among the DAT scenes, three scenarios: *citystreet*, *downtown*, and *lake* are directly derived from real-world locations with the digital twins tool. Specifically, the *citystreet* scenario is based on a small town in Los Angeles, the *downtown* scenario is derived from Manhattan, and the *lake* scenario is modeled after Wolf Lake Memorial Park in Indiana. In contrast, the *village*, *desert*, and *farmland* maps possess complex and unique features that are not adequately captured by OpenStreetMap (OSM) data. For example, the *village* map features mountainous terrain with tunnels, while the *farmland* map is characterized by diverse multicolored patterns. To overcome these limitations, we use Creo software [29] to model detailed scene elements, which are then integrated into the webots for constructing realistic maps.

Targets. All tracking target illustrations are presented in Fig. 9. Specifically, Fig. 9(a) presents *automobile* and *motorbike* tracking targets, including passenger vehicles (the first seven cars), buses, trucks, trailers, and motorcycles (such as scooters and motorbikes). These two categories of tracking targets leverage Simulation of Urban Mobility (SUMO) [36] for road behavior modeling and interaction management with other targets. In contrast, Fig. 9(b)-(d) display *pedestrian*, *wheeled robot*, and *legged robot* tracking targets, respectively. These three types of targets utilize SUMO paths for position initialization and rely on specific controllers for action and behavior management.

Sensors. In the real world VAT tasks, a single camera cannot ensure the agent's stability and robustness. Thus, integration with other sensors is often required. The DAT benchmark provides common sensors that can obtain the drone's state parameters relative to the world coordinate system. The drone's position and velocity are determined using GPS, while its acceleration is measured by an accelerometer, providing essential self-referential data for visual navigation tasks. Angular velocity is recorded via a gyroscope, and Euler angles obtained from the IMU are converted into quaternions to facilitate state estimation and ensure orientation stability. Additionally, the *RPLIDAR A2*, provided by DAT, generates point cloud data, which supports tasks such as obstacle avoidance and navigation

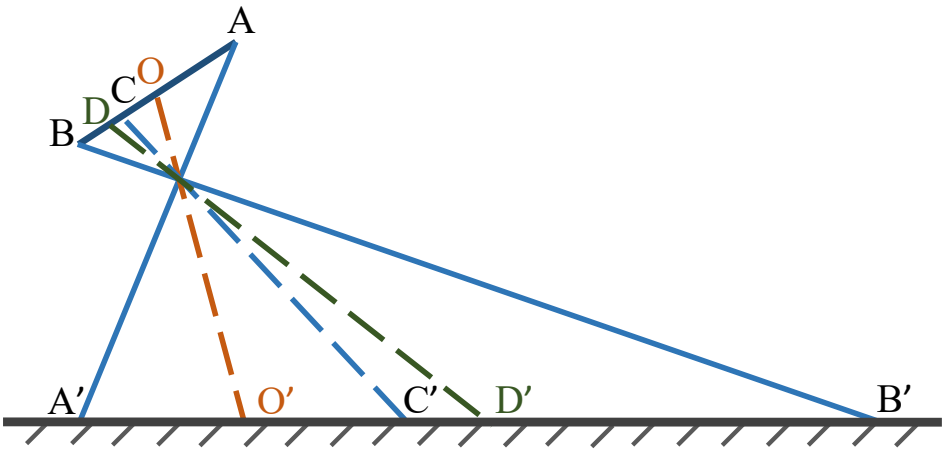


Figure 10: Diagram for the theoretical proof of **Proposition 1**.

by delivering detailed environmental information. The specific sensors, parameters and potential tasks are in Table 8.

Additional Parameters. The training process of VAT agents often requires additional parameters for effective reward design. To facilitate this, DAT benchmark provides 4 categories comprising a total of 13 parameters, supporting diverse reward design strategies, as detailed in Table 9.

First are the camera parameters, which mainly include image width `cameraWidth`, image height `cameraHeight`, field of view `cameraFov`, and focal length `cameraF`. Utilizing these, the camera plane can be projected onto the ground to aid in reward construction.

Next is the homogeneous transformation matrix (HTM). In the reward design, coordinate transformations are often required to express physical quantities within a unified coordinate system, enabling consistent calculations. For example, prior studies [19, 39, 18] transform the position, velocity, and acceleration of targets into the tracker's coordinate system to construct rewards. To support such operations, DAT benchmark provides $\mathtt{T}_{\mathtt{cw}}$, the HTM mapping the drone camera coordinate system to the world coordinate system, and $\mathtt{T}_{\mathtt{tw}}$, the HTM mapping the tracking target's coordinate system to the world coordinate system.

Additionally, for the state of the tracker itself, `cameraMidPos` represents the position of the drone camera's optical center in the world coordinate system. The parameter `crash` indicates whether the drone collides with any buildings in the scene, which can be used in reward design for obstacle avoidance tasks.

Lastly, for ease of model training in simulations, reward design often depends on some privileged information, i.e., variables that are almost impossible to obtain in real-world settings. Thus, DAT benchmark also provides such adaptations. For example, `carMidGlobalPos` gives the target's position in the world coordinate system, and `carDronePosOri` represents the target's orientation and position relative to the drone coordinate system, frequently used in VAT reward design [19, 39, 18]. Furthermore, information on the target's direction and type is provided.

Task Configuration. We encapsulate the scenes, tasks, and domain randomization into Python classes, and provide 3 different environment classes for different algorithm requirements. The base environment class directly interacts with webots and is designed to support asynchronous reinforcement learning algorithms, such as the asynchronous advantage actor-critic (A3C) algorithm [42]. The Gymnasium environment class wraps the base environment class into a Gymnasium [60] interface, enabling direct compatibility with popular reinforcement learning libraries, such as Stable-Baselines3 [49] and Tianshou [68] for efficient algorithm development and evaluation. The parallel environment class encapsulates the base environment class to enable parallel execution, providing direct support for synchronous algorithms, such as proximal policy optimization (PPO) [55] and soft actor-critic (SAC) [25]. Additionally, the scenario selection, tracker and target configuration, SUMO parameters, task additional parameters, and randomization methods can all be efficiently customized through a JSON configuration file.

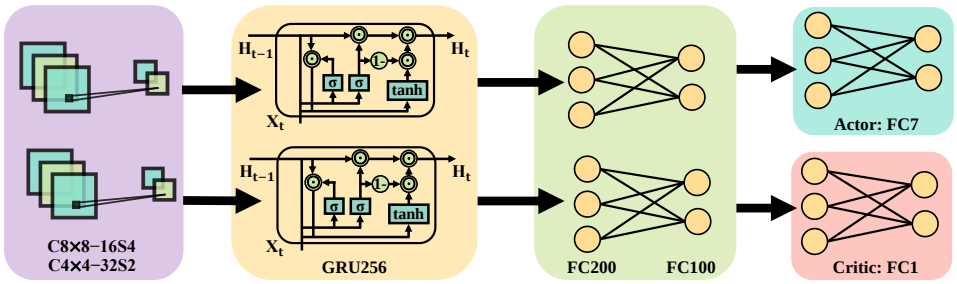


Figure 11: Network structure of Drone Agent.

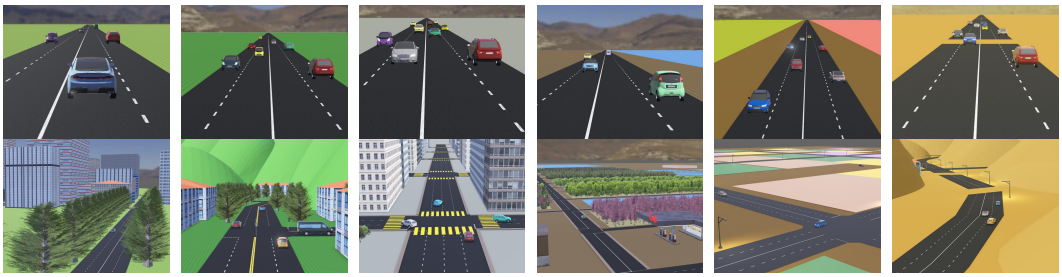


Figure 12: Schematic diagram of the training environments for the two-stage of Curriculum Learning

C More Details of Proposed GC-VAT

C.1 Theoretical Proof of Reward Design

Theoretical proof of Proposition 1. Consider two points A and B on the image symmetry axis (in Fig. 10), which are symmetric with respect to the image center O. The projections of these points onto the ground are denoted as A', B' and O', respectively. Take a point C' on line segment $O'B'$ such that the Euclidean distance $d(O', C') = d(A', O')$.

Given:

1. In the image plane, the deviation $\phi(\cdot, \cdot)$ of point A and B from the image center O is the same, i.e. $\phi(A, O) = \phi(B, O)$.

2. In the projection plane, the Euclidean distance from A' and C' to the ideal position O' are equal, i.e. $d(A', O') = d(O', C')$.

It is evident that:

1. For any point D' on line segment $B'C'$, the following relationship holds: $d(O', D') > d(A', O')$.

2. The corresponding point D in the image lies on line segment BC, and thus $\phi(D, O) < \phi(A, O)$.

Thus, it is clear that the actual distance between the target and the ideal position is inconsistent with the deviation of the target from the image center in the image.

Theoretical proof of Remark 1. According to **Proposition 1**, the following relationship between the Euclidean distance and the deviation holds:

$$\exists P_1, P_2 \in \mathcal{I}_p, \text{s.t. } \phi_1 < \phi_2, d_1 > d_2, \tag{8}$$

where $\phi_i = \phi(P_i, C_g)$ and $d_i = d(P_i, C_g)$. Therefore, for a distance-based reward function $\mathcal{R}_d(\cdot)$ that satisfies the **Reward Design Principle**, it follows that:

$$\exists P_1, P_2 \in \mathcal{I}_p, \text{s.t. } \phi_1 < \phi_2, \mathcal{R}_d(d_1) < \mathcal{R}_d(d_2). \tag{9}$$

Table 10: Total training steps on different scenes. During the training process, we employ a parallel training approach involving 35 agents. Consequently, the reported total training steps represent the cumulative steps taken by all agents combined.

Scene	citystreet	desert	village	downtown	lake	farmland
Steps (M)	19.2	13.4	21.3	19.8	9.9	9.2

Table 11: Transition steps across different scenes.

Scene	citystreet	desert	village	downtown	lake	farmland
T (M)	10.0	6.2	8.0	10.3	5.6	4.1

C.2 More Details

Network Structure. The structure of the GC-VAT is shown in Fig. 11. In this figure, C8×8-16S4 represents 16 convolutional filters of size 8×8 and stride 4. GRU256 denotes a GRU network with 256 hidden units, and FC200 represents a fully connected layer with 200 neurons.

Domain Randomization. While simpler settings facilitate the agent's learning of task objectives, they also heighten the risk of the agent rapidly converging to a suboptimal action distribution, undermining the exploration process. Consequently, implementing domain randomization is essential. This is achieved through the randomization of the drone's initial position and orientation relative to the target, necessitating a broader range of actions to maximize rewards. Moreover, to enhance the agent's spatial perception ability, randomization is also introduced in its gimbal pitch angle.

In our two-stage curriculum learning process, we employ identical domain randomization. The flight altitude is selected from the interval $[13, 22]$m, and the camera pitch angle is chosen from $[0.6, 1.38]$rad. These parameters are consistent throughout each episode. Meanwhile, the drone's initial orientation relative to the target fluctuates within the range $[-\pi, \pi]$rad, and the target's initial position is set between $[-4.5, -2.5] \cup [2.5, 4.5]$m.

Details on coordinate transformations. Given two planes $P_0 : \hat{n_0}\mathbf{x}^T + D_0 = 0$ and $P_1 : \hat{n_1}\mathbf{x}^T + D_1 = 0$, along with the HTM T_{01} from P_0 to P_1. The T_{01} is defined as:

$$T_{01} = \begin{bmatrix} R_{01} & t_{01} \\ 0 & 1 \end{bmatrix}. \tag{10}$$

Hence, the expression of plane P_1 can be obtained using the analytical expression of plane P_0 and T_{01} as follows:

$$\begin{aligned} \hat{n_1}^T &= R_{01}\hat{n_0}^T, \\ D_1 &= D_0 - \hat{n_1}t_{01}. \end{aligned} \tag{11}$$

Considering the ground plane $G_w : z = h$ in the world coordinate system $\{\mathrm{w}\}$, with representation in the camera coordinate system $\{\mathrm{c}\}$ denoted as $G_c : A_g x + B_g y + C_g z + D_g = 0$, the vectors of these two planes are $P_{G_w} = (0, 0, 1, -h)$ and $P_{G_c} = (A_g, B_g, C_g, D_g)$.

Furthermore, from Table 9, we can obtain the HTM $\mathtt{T}_{cw}$ from $\{\mathrm{c}\}$ to $\{\mathrm{w}\}$ defined as follows:

$$T_{cw} = \begin{bmatrix} R_{cw} & t_{cw} \\ 0 & 1 \end{bmatrix}, \tag{12}$$

where R_{cw} is the rotation matrix from $\{\mathrm{c}\}$ to $\{\mathrm{w}\}$, which can be expressed in row vector form as: $R_{cw} = [r_1, r_2, r_3]^T$. Therefore, the homogeneous transformation matrix (HTM) T_{wc}, which represents the transformation from the world coordinate system $\{w\}$ to the camera coordinate system $\{c\}$, can be expressed as follows:

$$T_{wc} = \begin{bmatrix} R_{cw}^T & -R_{cw}^T t_{cw} \\ 0 & 1 \end{bmatrix}. \tag{13}$$

Using Eq. 11 and the matrix T_{wc}, the plane G_c can be formulated as $P_{G_c} = (r_3^T, -h + r_3^T R_{cw}^T t_{cw})$.

Privileged knowledge available for Drone Agent. During training in the simulator, the drone agent has access to additional information (e.g., the precise location of the target). However, during testing and real-world deployment, such privileged knowledge is **not** available.

Table 12: The detailed results of comparison experiments on CR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$49_{\pm 3}$	$49_{\pm 9}$	$45_{\pm 5}$	$49_{\pm 3}$	$48_{\pm 3}$	$48_{\pm 3}$	$49_{\pm 4}$	$49_{\pm 3}$	$49_{\pm 3}$
D-VAT	$48_{\pm 8}$	$46_{\pm 12}$	$46_{\pm 10}$	$57_{\pm 11}$	$50_{\pm 8}$	$46_{\pm 3}$	$48_{\pm 9}$	$54_{\pm 10}$	$53_{\pm 10}$
GC-VAT	**279**$_{\pm 110}$	**129**$_{\pm 112}$	**153**$_{\pm 119}$	**135**$_{\pm 109}$	**112**$_{\pm 92}$	**191**$_{\pm 122}$	**257**$_{\pm 126}$	**316**$_{\pm 84}$	**202**$_{\pm 119}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$9_{\pm 0}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 0}$	$9_{\pm 0}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$
D-VAT	$51_{\pm 10}$	$47_{\pm 13}$	$46_{\pm 10}$	$56_{\pm 11}$	$39_{\pm 8}$	$47_{\pm 3}$	$48_{\pm 13}$	$48_{\pm 13}$	$39_{\pm 10}$
GC-VAT	**278**$_{\pm 111}$	**307**$_{\pm 124}$	**305**$_{\pm 94}$	**119**$_{\pm 110}$	**170**$_{\pm 139}$	**275**$_{\pm 121}$	**182**$_{\pm 131}$	**307**$_{\pm 124}$	**307**$_{\pm 97}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$51_{\pm 7}$	$51_{\pm 11}$	$46_{\pm 5}$	$49_{\pm 4}$	$52_{\pm 11}$	$57_{\pm 24}$	$47_{\pm 5}$	$47_{\pm 5}$	$47_{\pm 5}$
D-VAT	$46_{\pm 8}$	$45_{\pm 9}$	$44_{\pm 8}$	$69_{\pm 42}$	$45_{\pm 8}$	$45_{\pm 3}$	$44_{\pm 8}$	$44_{\pm 8}$	$43_{\pm 8}$
GC-VAT	**234**$_{\pm 122}$	**160**$_{\pm 139}$	**239**$_{\pm 134}$	**93**$_{\pm 102}$	**153**$_{\pm 115}$	**140**$_{\pm 118}$	**257**$_{\pm 122}$	**257**$_{\pm 120}$	**114**$_{\pm 115}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$52_{\pm 3}$	$52_{\pm 9}$	$48_{\pm 7}$	$54_{\pm 5}$	$53_{\pm 5}$	$54_{\pm 8}$	$54_{\pm 5}$	$54_{\pm 5}$	$54_{\pm 5}$
D-VAT	$8_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 2}$
GC-VAT	**209**$_{\pm 131}$	**184**$_{\pm 136}$	**202**$_{\pm 129}$	**203**$_{\pm 119}$	**189**$_{\pm 93}$	**223**$_{\pm 114}$	**167**$_{\pm 135}$	**165**$_{\pm 126}$	**178**$_{\pm 125}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$49_{\pm 3}$	$49_{\pm 10}$	$46_{\pm 5}$	$49_{\pm 3}$	$47_{\pm 3}$	$49_{\pm 3}$	$48_{\pm 3}$	$48_{\pm 4}$	$48_{\pm 3}$
D-VAT	$50_{\pm 8}$	$45_{\pm 9}$	$45_{\pm 10}$	$70_{\pm 42}$	$46_{\pm 8}$	$43_{\pm 2}$	$46_{\pm 8}$	$51_{\pm 8}$	$49_{\pm 9}$
GC-VAT	**112**$_{\pm 86}$	**144**$_{\pm 110}$	**203**$_{\pm 133}$	**143**$_{\pm 134}$	**181**$_{\pm 116}$	**214**$_{\pm 111}$	**190**$_{\pm 129}$	**168**$_{\pm 110}$	**99**$_{\pm 67}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$51_{\pm 7}$	$50_{\pm 9}$	$46_{\pm 5}$	$49_{\pm 3}$	$51_{\pm 9}$	$60_{\pm 25}$	$48_{\pm 4}$	$56_{\pm 24}$	$56_{\pm 24}$
D-VAT	$13_{\pm 2}$	$13_{\pm 1}$	$13_{\pm 1}$	$15_{\pm 1}$	$14_{\pm 1}$	$13_{\pm 1}$	$14_{\pm 1}$	$13_{\pm 1}$	$14_{\pm 1}$
GC-VAT	**162**$_{\pm 89}$	**170**$_{\pm 125}$	**237**$_{\pm 128}$	**81**$_{\pm 71}$	**159**$_{\pm 119}$	**243**$_{\pm 117}$	**253**$_{\pm 109}$	**245**$_{\pm 117}$	**168**$_{\pm 105}$

Table 13: The detailed results of comparison experiments on TSR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.25_{\pm 0.02}$	$0.24_{\pm 0.03}$	$0.22_{\pm 0.03}$	$0.25_{\pm 0.02}$	$0.23_{\pm 0.03}$	$0.24_{\pm 0.01}$	$0.25_{\pm 0.02}$	$0.25_{\pm 0.02}$	$0.24_{\pm 0.02}$
D-VAT	$0.26_{\pm 0.02}$	$0.25_{\pm 0.04}$	$0.25_{\pm 0.02}$	$0.32_{\pm 0.08}$	$0.27_{\pm 0.04}$	$0.19_{\pm 0.01}$	$0.26_{\pm 0.02}$	$0.28_{\pm 0.02}$	$0.29_{\pm 0.02}$
GC-VAT	**0.80**$_{\pm 0.30}$	**0.54**$_{\pm 0.32}$	**0.50**$_{\pm 0.32}$	**0.45**$_{\pm 0.30}$	**0.44**$_{\pm 0.24}$	**0.66**$_{\pm 0.27}$	**0.72**$_{\pm 0.29}$	**0.93**$_{\pm 0.14}$	**0.79**$_{\pm 0.24}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.01}$
D-VAT	$0.27_{\pm 0.02}$	$0.26_{\pm 0.04}$	$0.25_{\pm 0.02}$	$0.32_{\pm 0.07}$	$0.23_{\pm 0.03}$	$0.26_{\pm 0.01}$	$0.26_{\pm 0.04}$	$0.26_{\pm 0.04}$	$0.26_{\pm 0.04}$
GC-VAT	**0.73**$_{\pm 0.31}$	**0.84**$_{\pm 0.29}$	**0.87**$_{\pm 0.19}$	**0.38**$_{\pm 0.32}$	**0.56**$_{\pm 0.28}$	**0.82**$_{\pm 0.25}$	**0.57**$_{\pm 0.31}$	**0.86**$_{\pm 0.28}$	**0.86**$_{\pm 0.22}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.25_{\pm 0.03}$	$0.25_{\pm 0.04}$	$0.23_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.25_{\pm 0.02}$	$0.26_{\pm 0.06}$	$0.23_{\pm 0.03}$	$0.23_{\pm 0.03}$	$0.23_{\pm 0.03}$
D-VAT	$0.23_{\pm 0.04}$	$0.23_{\pm 0.04}$	$0.22_{\pm 0.05}$	$0.31_{\pm 0.14}$	$0.24_{\pm 0.06}$	$0.22_{\pm 0.01}$	$0.22_{\pm 0.04}$	$0.22_{\pm 0.05}$	$0.23_{\pm 0.05}$
GC-VAT	**0.72**$_{\pm 0.28}$	**0.51**$_{\pm 0.34}$	**0.73**$_{\pm 0.32}$	**0.46**$_{\pm 0.29}$	**0.59**$_{\pm 0.33}$	**0.48**$_{\pm 0.31}$	**0.71**$_{\pm 0.32}$	**0.71**$_{\pm 0.32}$	**0.40**$_{\pm 0.29}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.30_{\pm 0.04}$	$0.26_{\pm 0.05}$	$0.27_{\pm 0.02}$	$0.29_{\pm 0.01}$	$0.29_{\pm 0.03}$	$0.29_{\pm 0.02}$	$0.29_{\pm 0.01}$	$0.29_{\pm 0.01}$	$0.29_{\pm 0.01}$
D-VAT	$0.05_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$
GC-VAT	**0.77**$_{\pm 0.31}$	**0.65**$_{\pm 0.30}$	**0.67**$_{\pm 0.29}$	**0.65**$_{\pm 0.30}$	**0.49**$_{\pm 0.29}$	**0.63**$_{\pm 0.33}$	**0.58**$_{\pm 0.31}$	**0.65**$_{\pm 0.29}$	**0.64**$_{\pm 0.28}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.25_{\pm 0.02}$	$0.25_{\pm 0.03}$	$0.23_{\pm 0.03}$	$0.24_{\pm 0.02}$	$0.24_{\pm 0.02}$	$0.24_{\pm 0.01}$	$0.24_{\pm 0.01}$	$0.24_{\pm 0.02}$	$0.24_{\pm 0.01}$
D-VAT	$0.25_{\pm 0.04}$	$0.23_{\pm 0.04}$	$0.23_{\pm 0.05}$	$0.30_{\pm 0.15}$	$0.26_{\pm 0.06}$	$0.22_{\pm 0.01}$	$0.26_{\pm 0.06}$	$0.26_{\pm 0.06}$	$0.25_{\pm 0.06}$
GC-VAT	**0.43**$_{\pm 0.25}$	**0.47**$_{\pm 0.30}$	**0.64**$_{\pm 0.31}$	**0.43**$_{\pm 0.28}$	**0.61**$_{\pm 0.31}$	**0.59**$_{\pm 0.30}$	**0.59**$_{\pm 0.39}$	**0.62**$_{\pm 0.32}$	**0.41**$_{\pm 0.24}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
AOT	$0.24_{\pm 0.02}$	$0.24_{\pm 0.04}$	$0.22_{\pm 0.03}$	$0.25_{\pm 0.02}$	$0.24_{\pm 0.02}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$
D-VAT	$0.07_{\pm 0.01}$	$0.07_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.08_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.08_{\pm 0.00}$
GC-VAT	**0.48**$_{\pm 0.24}$	**0.59**$_{\pm 0.34}$	**0.72**$_{\pm 0.26}$	**0.33**$_{\pm 0.20}$	**0.58**$_{\pm 0.28}$	**0.68**$_{\pm 0.32}$	**0.67**$_{\pm 0.32}$	**0.78**$_{\pm 0.22}$	**0.51**$_{\pm 0.28}$

Sparse Reward. In addition to the dense reward function described in the main text, we also provide a sparse reward function design. The sparse reward only provides a fixed reward when the target is within the image and no reward when it is outside. The definition of r_d is as follows.

$$r_d = \begin{cases} 1, & t \in \mathcal{I} \\ 0, & \text{otherwise} \end{cases}, \tag{14}$$

where $\mathcal{I}$ represents the image range. This reward can be used to construct the metric, Tracking Success Rate (TSR).

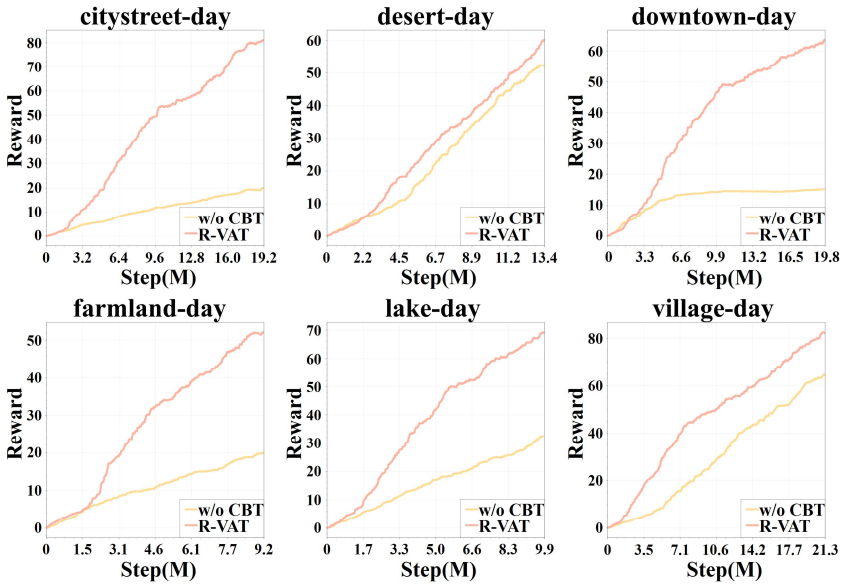


Figure 13: Schematic diagram of reward curves on DAT scenes.

Training algorithm. For the training method of GC-VAT, we choose to use PPO algorithm. PPO algorithm regulates the speed of gradient updates by constraining the magnitude of policy changes r_t, expressed as follows:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, \tag{15}$$

where π_θ and $\pi_{\theta_{old}}$ are the new and old policies. Additionally, to enhance the agent's exploration, we introduce an entropy loss term $\mathcal{H}$, formulated as:

$$\mathcal{H}(\pi_\theta(s)) = -\sum_a \pi_\theta(a|s)\log\pi_\theta(a|s). \tag{16}$$

The optimization objective for the actor is as follows:

$$\mathcal{L}_A = \hat{\mathbb{E}}[\min(r_t\hat{A}_t, \text{clip}(r_t, 1-\epsilon, 1+\epsilon)\hat{A}_t) + \beta\mathcal{H}], \tag{17}$$

where $\hat{A}_t$ is the advantage function, ϵ is the clip parameter, and β is the entropy coefficient. The expression of $\hat{A}_t$ is:

$$\hat{A}_t = \sum_{l=0}^{E_l - t} (\gamma\lambda)^l \delta_{t+l}, \tag{18}$$

where T, λ, δ_{t+l} are the data collection step, generalized advantage estimator (GAE) [56] discount factor and temporal difference error respectively. The optimization objective expression of the critic network V is defined as:

$$\mathcal{L}_C = \hat{\mathbb{E}}_t[(r_t + \gamma V(s_{t+1}) - V(s_t))^2]. \tag{19}$$

The hyperparameters of the PPO algorithm used in this article are set as follows: discount factor $\gamma = 0.9$, GAE discount factor $\lambda = 0.95$, entropy coefficient $\beta = 0.01$, PPO clipping parameter $\epsilon = 0.2$.

Curriculum Learning for Agent Training. We introduce a Curriculum-Based Training (CBT) strategy designed to progressively enhance the performance of the tracker. In the first-stage curriculum, the agent is trained to track vehicles moving along straight trajectories without occlusions or extra interference. In the second-stage curriculum, the agent is exposed to visually complex environments and tasked with tracking targets exhibiting diverse and dynamic behaviors. The scenario of each stage is shown in Fig. 12, where the upper row is the first-stage environment, and the lower row corresponds to the second-stage environment.

Table 14: Effectiveness of CBT strategy on the DAT benchmark, results from CR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$54_{\pm 7}$	$37_{\pm 21}$	$30_{\pm 6}$	$30_{\pm 14}$	$48_{\pm 13}$	$48_{\pm 4}$	$54_{\pm 9}$	$54_{\pm 9}$	$54_{\pm 9}$
GC-VAT	**279**$_{\pm 110}$	**129**$_{\pm 112}$	**153**$_{\pm 119}$	**135**$_{\pm 109}$	**112**$_{\pm 92}$	**191**$_{\pm 122}$	**257**$_{\pm 126}$	**316**$_{\pm 84}$	**202**$_{\pm 119}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$253_{\pm 132}$	$302_{\pm 99}$	$284_{\pm 92}$	**175**$_{\pm 102}$	**236**$_{\pm 123}$	$266_{\pm 110}$	**241**$_{\pm 127}$	$279_{\pm 120}$	$306_{\pm 95}$
GC-VAT	**278**$_{\pm 111}$	**307**$_{\pm 124}$	**305**$_{\pm 94}$	$119_{\pm 110}$	$170_{\pm 139}$	**275**$_{\pm 121}$	$182_{\pm 131}$	**307**$_{\pm 124}$	**307**$_{\pm 97}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$230_{\pm 120}$	**197**$_{\pm 124}$	**255**$_{\pm 118}$	$59_{\pm 69}$	$126_{\pm 105}$	**182**$_{\pm 120}$	**267**$_{\pm 93}$	$208_{\pm 141}$	$73_{\pm 68}$
GC-VAT	**234**$_{\pm 122}$	$160_{\pm 139}$	$239_{\pm 134}$	**93**$_{\pm 102}$	**153**$_{\pm 115}$	$140_{\pm 118}$	$257_{\pm 122}$	**257**$_{\pm 120}$	**114**$_{\pm 115}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$54_{\pm 9}$	$49_{\pm 13}$	$47_{\pm 8}$	$57_{\pm 15}$	$51_{\pm 9}$	$48_{\pm 4}$	$29_{\pm 3}$	$57_{\pm 15}$	$58_{\pm 15}$
GC-VAT	**209**$_{\pm 131}$	**184**$_{\pm 136}$	**202**$_{\pm 129}$	**203**$_{\pm 119}$	**189**$_{\pm 93}$	**223**$_{\pm 114}$	**167**$_{\pm 135}$	**165**$_{\pm 126}$	**178**$_{\pm 125}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	**124**$_{\pm 90}$	$88_{\pm 52}$	$191_{\pm 108}$	$93_{\pm 75}$	**187**$_{\pm 123}$	$198_{\pm 117}$	$183_{\pm 110}$	**185**$_{\pm 102}$	**102**$_{\pm 57}$
GC-VAT	$112_{\pm 86}$	**144**$_{\pm 110}$	**203**$_{\pm 133}$	**143**$_{\pm 134}$	$181_{\pm 116}$	**214**$_{\pm 111}$	**190**$_{\pm 129}$	$168_{\pm 110}$	$99_{\pm 67}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$52_{\pm 9}$	$47_{\pm 9}$	$45_{\pm 9}$	$69_{\pm 42}$	$50_{\pm 9}$	$46_{\pm 2}$	$46_{\pm 2}$	$46_{\pm 3}$	$46_{\pm 2}$
GC-VAT	**162**$_{\pm 89}$	**170**$_{\pm 125}$	**237**$_{\pm 128}$	**81**$_{\pm 71}$	**159**$_{\pm 119}$	**243**$_{\pm 117}$	**253**$_{\pm 109}$	**245**$_{\pm 117}$	**168**$_{\pm 105}$

Table 15: Effectiveness of CBT strategy on the DAT benchmark, results from TSR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$0.30_{\pm 0.05}$	$0.14_{\pm 0.10}$	$0.20_{\pm 0.10}$	$0.31_{\pm 0.15}$	$0.28_{\pm 0.06}$	$0.21_{\pm 0.01}$	$0.30_{\pm 0.05}$	$0.30_{\pm 0.05}$	$0.30_{\pm 0.05}$
GC-VAT	**0.80**$_{\pm 0.30}$	**0.54**$_{\pm 0.32}$	**0.50**$_{\pm 0.32}$	**0.45**$_{\pm 0.30}$	**0.44**$_{\pm 0.24}$	**0.66**$_{\pm 0.27}$	**0.72**$_{\pm 0.29}$	**0.93**$_{\pm 0.14}$	**0.79**$_{\pm 0.24}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	**0.83**$_{\pm 0.28}$	$0.75_{\pm 0.32}$	$0.66_{\pm 0.34}$	**0.52**$_{\pm 0.28}$	**0.69**$_{\pm 0.24}$	$0.74_{\pm 0.26}$	**0.59**$_{\pm 0.36}$	$0.74_{\pm 0.34}$	$0.75_{\pm 0.34}$
GC-VAT	$0.73_{\pm 0.31}$	**0.84**$_{\pm 0.29}$	**0.87**$_{\pm 0.19}$	$0.38_{\pm 0.32}$	$0.56_{\pm 0.28}$	**0.82**$_{\pm 0.25}$	$0.57_{\pm 0.31}$	**0.86**$_{\pm 0.28}$	**0.86**$_{\pm 0.22}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	**0.73**$_{\pm 0.28}$	**0.62**$_{\pm 0.28}$	**0.82**$_{\pm 0.16}$	$0.23_{\pm 0.17}$	$0.46_{\pm 0.25}$	**0.58**$_{\pm 0.33}$	**0.71**$_{\pm 0.28}$	$0.69_{\pm 0.33}$	**0.40**$_{\pm 0.24}$
GC-VAT	$0.72_{\pm 0.28}$	$0.51_{\pm 0.34}$	$0.73_{\pm 0.32}$	**0.46**$_{\pm 0.29}$	**0.59**$_{\pm 0.33}$	$0.48_{\pm 0.31}$	$0.71_{\pm 0.32}$	**0.71**$_{\pm 0.32}$	$0.40_{\pm 0.29}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$0.29_{\pm 0.04}$	$0.27_{\pm 0.03}$	$0.27_{\pm 0.03}$	$0.33_{\pm 0.06}$	$0.28_{\pm 0.03}$	$0.27_{\pm 0.01}$	$0.33_{\pm 0.06}$	$0.33_{\pm 0.06}$	$0.33_{\pm 0.06}$
GC-VAT	**0.77**$_{\pm 0.31}$	**0.65**$_{\pm 0.30}$	**0.67**$_{\pm 0.29}$	**0.65**$_{\pm 0.30}$	**0.49**$_{\pm 0.29}$	**0.63**$_{\pm 0.33}$	**0.58**$_{\pm 0.31}$	**0.65**$_{\pm 0.29}$	**0.64**$_{\pm 0.28}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	**0.51**$_{\pm 0.30}$	**0.47**$_{\pm 0.29}$	$0.45_{\pm 0.22}$	**0.44**$_{\pm 0.23}$	$0.57_{\pm 0.28}$	$0.59_{\pm 0.26}$	**0.78**$_{\pm 0.22}$	**0.62**$_{\pm 0.24}$	$0.33_{\pm 0.15}$
GC-VAT	$0.43_{\pm 0.25}$	$0.47_{\pm 0.30}$	**0.64**$_{\pm 0.31}$	$0.43_{\pm 0.28}$	**0.61**$_{\pm 0.31}$	**0.59**$_{\pm 0.30}$	$0.59_{\pm 0.39}$	$0.62_{\pm 0.32}$	**0.41**$_{\pm 0.24}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
w/o CBT	$0.26_{\pm 0.04}$	$0.24_{\pm 0.04}$	$0.23_{\pm 0.05}$	$0.31_{\pm 0.14}$	$0.26_{\pm 0.06}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$	$0.23_{\pm 0.01}$
GC-VAT	**0.48**$_{\pm 0.24}$	**0.59**$_{\pm 0.34}$	**0.72**$_{\pm 0.26}$	**0.33**$_{\pm 0.20}$	**0.58**$_{\pm 0.28}$	**0.68**$_{\pm 0.32}$	**0.67**$_{\pm 0.32}$	**0.78**$_{\pm 0.22}$	**0.51**$_{\pm 0.28}$

D Baselines

Active Object Tracking (AOT) [39]. In this paper, the agent learns to follow a fixed target-tracking trajectory using A3C. In addition, the agent uses the following reward:

$$r = A - (\frac{\sqrt{x^2 + (y-d)^2}}{c} + \lambda \mid \omega \mid), \tag{20}$$

where d represents the optimal distance between the tracker and the target, c is the maximum allowable distance, and A denotes the maximum reward. In the original paper, $c = 200$ and $A = 1.0$. During our replication, we set $A = 1.0$, but due to the drone's camera being tilted downward, a value of $c = 200$ would far exceed the camera's field of view, which is unrealistic. Therefore, we modify the parameter c to be the maximum offset distance that keeps the target within the image, i.e., $c = 9$.

D-VAT[19]. In this approach, the agent uses an asymmetric Actor-Critic network structure and the soft actor-critic learning method [25] to accomplish the task of drone tracking another drone. In the

Table 16: Effectiveness of reward design on the DAT benchmark, results from CR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$9_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 0}$	$8_{\pm 1}$	$9_{\pm 0}$	$9_{\pm 0}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$
GC-VAT	**279**$_{\pm 110}$	**129**$_{\pm 112}$	**153**$_{\pm 119}$	**135**$_{\pm 109}$	**112**$_{\pm 92}$	**191**$_{\pm 122}$	**257**$_{\pm 126}$	**316**$_{\pm 84}$	**202**$_{\pm 119}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$9_{\pm 1}$	$9_{\pm 0}$	$8_{\pm 1}$	$9_{\pm 0}$	$8_{\pm 0}$	$10_{\pm 0}$	$8_{\pm 1}$	$10_{\pm 1}$	$8_{\pm 0}$
GC-VAT	**278**$_{\pm 111}$	**307**$_{\pm 124}$	**305**$_{\pm 94}$	**119**$_{\pm 110}$	**170**$_{\pm 139}$	**275**$_{\pm 121}$	**182**$_{\pm 131}$	**307**$_{\pm 124}$	**307**$_{\pm 97}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$9_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 0}$	$8_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 1}$
GC-VAT	**234**$_{\pm 122}$	**160**$_{\pm 139}$	**239**$_{\pm 134}$	**93**$_{\pm 102}$	**153**$_{\pm 115}$	**140**$_{\pm 118}$	**257**$_{\pm 122}$	**257**$_{\pm 120}$	**114**$_{\pm 115}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$8_{\pm 1}$	$8_{\pm 0}$	$8_{\pm 1}$	$9_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 0}$
GC-VAT	**209**$_{\pm 131}$	**184**$_{\pm 136}$	**202**$_{\pm 129}$	**203**$_{\pm 119}$	**189**$_{\pm 93}$	**223**$_{\pm 114}$	**167**$_{\pm 135}$	**165**$_{\pm 126}$	**178**$_{\pm 125}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$11_{\pm 3}$	$11_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 2}$	$9_{\pm 0}$	$8_{\pm 0}$	$9_{\pm 0}$	$10_{\pm 1}$	$8_{\pm 1}$
GC-VAT	**112**$_{\pm 86}$	**144**$_{\pm 110}$	**203**$_{\pm 133}$	**143**$_{\pm 134}$	**181**$_{\pm 116}$	**214**$_{\pm 111}$	**190**$_{\pm 129}$	**168**$_{\pm 110}$	**99**$_{\pm 67}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$9_{\pm 1}$	$8_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$8_{\pm 1}$	$9_{\pm 1}$	$9_{\pm 0}$	$9_{\pm 0}$	$9_{\pm 0}$
GC-VAT	**162**$_{\pm 89}$	**170**$_{\pm 125}$	**237**$_{\pm 128}$	**81**$_{\pm 71}$	**159**$_{\pm 119}$	**243**$_{\pm 117}$	**253**$_{\pm 109}$	**245**$_{\pm 117}$	**168**$_{\pm 105}$

Table 17: Effectiveness of reward design on the DAT benchmark, results from TSR metric.

	Within / Cross Scene						Cross Domain		
Train: citystreet	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.06_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$
GC-VAT	**0.80**$_{\pm 0.30}$	**0.54**$_{\pm 0.32}$	**0.50**$_{\pm 0.32}$	**0.45**$_{\pm 0.30}$	**0.44**$_{\pm 0.24}$	**0.66**$_{\pm 0.27}$	**0.72**$_{\pm 0.29}$	**0.93**$_{\pm 0.14}$	**0.79**$_{\pm 0.24}$
Train: desert	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.10_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.06_{\pm 0.00}$
GC-VAT	**0.73**$_{\pm 0.31}$	**0.84**$_{\pm 0.29}$	**0.87**$_{\pm 0.19}$	**0.38**$_{\pm 0.32}$	$0.56_{\pm 0.28}$	**0.82**$_{\pm 0.25}$	**0.57**$_{\pm 0.31}$	**0.86**$_{\pm 0.28}$	**0.86**$_{\pm 0.22}$
Train: village	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$
GC-VAT	**0.72**$_{\pm 0.28}$	**0.51**$_{\pm 0.34}$	**0.73**$_{\pm 0.32}$	**0.46**$_{\pm 0.29}$	**0.59**$_{\pm 0.33}$	**0.48**$_{\pm 0.31}$	**0.71**$_{\pm 0.32}$	**0.71**$_{\pm 0.32}$	**0.40**$_{\pm 0.29}$
Train: downtown	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$
GC-VAT	**0.77**$_{\pm 0.31}$	**0.65**$_{\pm 0.30}$	**0.67**$_{\pm 0.29}$	**0.65**$_{\pm 0.30}$	**0.49**$_{\pm 0.29}$	**0.63**$_{\pm 0.33}$	**0.58**$_{\pm 0.31}$	**0.65**$_{\pm 0.29}$	**0.64**$_{\pm 0.28}$
Train: lake	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.10_{\pm 0.01}$	$0.09_{\pm 0.01}$	$0.07_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.08_{\pm 0.00}$	$0.06_{\pm 0.00}$
GC-VAT	**0.43**$_{\pm 0.25}$	**0.47**$_{\pm 0.30}$	**0.64**$_{\pm 0.31}$	**0.43**$_{\pm 0.28}$	**0.61**$_{\pm 0.31}$	**0.59**$_{\pm 0.30}$	**0.59**$_{\pm 0.39}$	**0.62**$_{\pm 0.32}$	**0.41**$_{\pm 0.24}$
Train: farmland	citystreet	desert	village	downtown	lake	farmland	night	foggy	snow
$R_{\text{D-VAT}}$	$0.06_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.01}$	$0.05_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$	$0.06_{\pm 0.00}$
GC-VAT	**0.48**$_{\pm 0.24}$	**0.59**$_{\pm 0.34}$	**0.72**$_{\pm 0.26}$	**0.33**$_{\pm 0.20}$	**0.58**$_{\pm 0.28}$	**0.68**$_{\pm 0.32}$	**0.67**$_{\pm 0.32}$	**0.78**$_{\pm 0.22}$	**0.51**$_{\pm 0.28}$

actual comparative experiments, we convert it from a continuous action space to a discrete action space, referring to [10]. Additionally, the method uses the following reward function.

$$r(k) = \begin{cases} r_e(k) - k_v r_v(k) - k_u r_u(k) & \|y(k)\| > d_m \\ -k_c & \text{otherwise,} \end{cases} \tag{21}$$

In the above equation Eq. 21, $r_v(k)$ and $r_u(k)$ are regularization terms for the drone's speed and output control, as shown in Eq. 22. For the discrete action space, the regularization term has a fixed value for a given action. This term only regularizes the linear velocity of the drone, which causes the drone to tend to perform rotational movements. Therefore, in the reproduction process, we set $k_v = 0$ and $k_u = 0$. Additionally, due to the unexpectedly large acceleration values obtained for the target relative to the tracker under the discrete action setting, we set the input acceleration of the critic

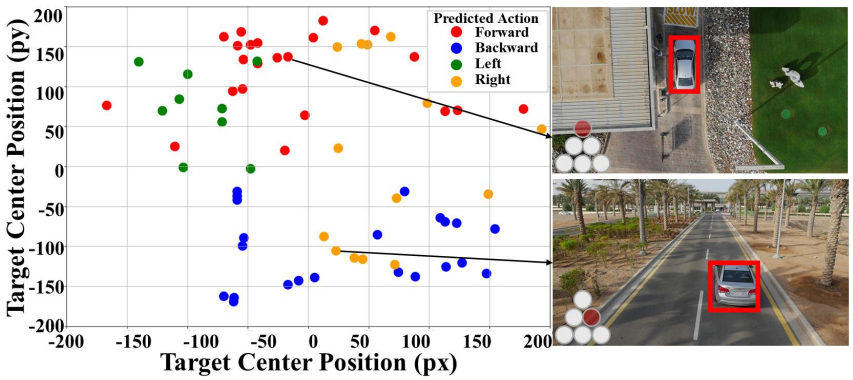


Figure 14: Qualitative results on images from the *car6* video sequence. Arrows link data points to the visualization of associated scenarios.

network to $a(k) = 0$.

$$r_v(k) = \frac{\|v(k)\|}{1 + \|v(k)\|}, \quad r_u(k) = \frac{\|u(k)\|}{1 + \|u(k)\|}. \tag{22}$$

It is important to note that in the AOT and D-VAT experiments, the target is initially positioned at the center of the tracker's image, and the initial forward directions of both the tracker and the target are aligned. Additionally, since the success criterion of DAT requires the agent to keep the target at the center of its view, the optimal distance between the tracker and the target is defined as the distance in the forward direction when the target is at the center of the camera's field of view. The tracker's flight altitude is set to 22 meters, and the gimbal pitch angle is 1.37 radians, which remains consistent with the parameters used during testing.

E More Experiments

E.1 Experiment Settings

More Implementation Details. The training involves a range of 9.2M to 21.3M steps across 35 parallel environments. The webots runs at 500Hz, with the algorithm updating every four steps (125Hz). Episodes last up to 1500 steps and were terminated early if the drone lost the target for over 100 consecutive steps, collided, or crashed. The drone translation speed is set to 40m/s, and rotational speed to 2rad/s. The map features 40 vehicles, each with a maximum speed of 20m/s and acceleration of $\pm 25\text{m/s}^2$. During testing, the altitude is set to 22m, the pitch angle to 1.37rad, and the target initializes at the camera's center.

The drone's translation speed is set to a higher value to prevent it from becoming too similar to the target's speed (with a maximum of 20 m/s). This prevents simple forward movement from yielding excessively high reward evaluations. If the drone's speed is set lower (e.g., 20 m/s), it may adopt a suboptimal strategy, relying solely on one action.

Due to the varying challenges posed by different scene maps, the convergence speed of the agent differs across experiments. The training steps are shown in Table 10.

Ablation Experiment Settings. In this section, we introduce the training conditions of the single-stage RL and GC-VAT, as well as the criteria for stage transitions. In single-stage RL, the agent is placed in one of six scenarios (*citystreet*, *desert*, *village*, *downtown*, *lake*, and *farmland*) for training. For GC-VAT, the agent is first trained in an environment where a randomly colored target moves straight along a line without obstacles. After convergence, the model is then trained in the corresponding complex scenarios. The transition steps **T** for GC-VAT are in Table 11.

E.2 Comparison Experiments

We provide a comprehensive analysis of the comparative experimental results.

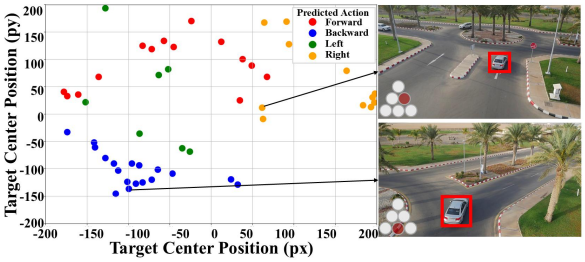


Figure 15: Qualitative results on the *car8* video.

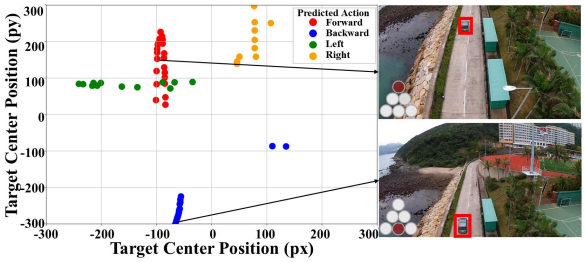


Figure 16: Qualitative results on the *Car4* video.

Table 18: Results (metric is Correct Action Rate) of 8 videos in VOT benchmark.

Video	*car1*	*car3*	*car6*	*car8*
Random	0.418	0.434	0.418	0.430
Ours	**0.696**	**0.845**	**0.754**	**0.833**
Video	*carchase*	*car16*	*following*	*car9*
Random	0.429	0.421	0.314	0.439
Ours	**0.870**	**0.834**	**0.773**	**0.756**

Table 19: Results of 8 videos in DTB70 benchmark.

Video	*Car2*	*Car4*	*Car5*	*RcCar4*
Random	0.419	0.421	0.429	0.436
Ours	**0.757**	**0.894**	**0.893**	**0.876**
Video	*Car8*	*RaceCar*	*RaceCar1*	*RcCar3*
Random	0.411	0.462	0.430	0.400
Ours	**0.803**	**0.713**	**0.880**	**0.851**

Specifically, we provide detailed evaluations for within-scene (same scenes, same weather), cross-scene and cross-domain testing. Table 12 reports the CR metric of three models under cross-scene and cross-domain conditions, while Table 13 presents the TSR metric.

As shown in Table 12 and Table 13, the proposed GC-VAT significantly outperforms SOTA methods. Due to the reward design based on physical distance, both the AOT [39] and D-VAT [19] fail to accurately reflect the agent's tracking performance from a top-down perspective (see Appendix C.1 for theoretical proof), leading to misleading training signals for the tracker. Consequently, neither AOT nor D-VAT can effectively learn meaningful features, resulting in irregular performance distributions. In contrast, the proposed GC-VAT achieves superior convergence across all scenes. Specifically, in cross-scene experiments, the testing performance of the agent on the *downtown* map is relatively low, indicating that dense buildings and complex road elements pose significant challenges to the agent. Conversely, the testing performance on the *village* map is comparatively high, suggesting that the uniform color and simpler road conditions in the village map present fewer challenges.

For cross-domain testing experiments, the agent performs well under *night* and *foggy* conditions but struggles under *snow* conditions. This indicates that the proposed GC-VAT exhibits strong robustness to changes in lighting and visibility but is less adaptive to variations in scene tone.

E.3 Ablation Experiments

We present a comprehensive analysis of the ablation studies. First, we provide the reward curves for the *citystreet*, *desert*, *village*, *downtown*, *lake*, and *farmland* maps (see Fig. 13). Next, we provide detailed experimental results on the effectiveness of the Curriculum-Based Training strategy, as shown in Table 14 and Table 15.

Finally, the effectiveness of the reward in the GC-VAT can be found in Table 16 and Table 17.

Effectiveness of reward design. To experimentally validate the effectiveness of the reward design proposed in this paper and to corroborate the theoretical proof in Appendix C.1, we conduct ablation experiments on the reward function. The comparative method utilizes the reward function from [19]. The detailed experimental results for the CR and TSR metrics are provided in Table 16 and Table 17. For within-scene testing, the GC-VAT achieves an average improvement of $1100\%(0.06 \rightarrow 0.72)$ in the TSR metric compared to the reward design in [19]. In cross-scene and cross-domain testing, the GC-VAT achieves average enhancements of $850\%(0.06 \rightarrow 0.57)$ and $1017\%(0.06 \rightarrow 0.67)$ in the TSR metric, respectively. These results demonstrate the high effectiveness of the proposed reward.

Effectiveness of Curriculum-Based Training strategy. To validate the effectiveness of the proposed Curriculum-Based Training (CBT) strategy, we conduct ablation experiments by removing the CBT module. The results for the CR and TSR metrics are presented in Table 14 and Table 15, respectively.

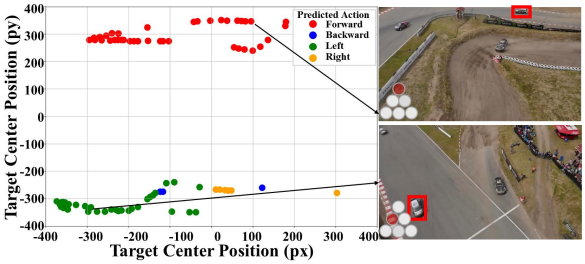


Figure 17: Qualitative results on the *RaceCar1* video.

Table 20: Results of 8 videos in UAVDT [20] benchmark.

Video	*S1603*	*S0201*	*S0101*	*S0306*
Random	0.435	0.438	0.422	0.437
Ours	**0.896**	**0.806**	**0.865**	**0.773**
Video	*S1201*	*S0303*	*S1301*	*S1701*
Random	0.445	0.385	0.397	0.407
Ours	**0.867**	**0.735**	**0.760**	**0.713**

The experimental results demonstrate that single-stage reinforcement learning methods without the CBT strategy successfully learn task objectives and achieve convergence on the *desert*, *village*, and *lake* maps. These three maps exhibit similar environmental characteristics: the *desert* and *village* maps feature uniform background colors and relatively simple road elements. Although the *desert* map has road segments partially covered by sand, these challenges are easy for the agent to overcome. Similarly, while the *village* map includes tunnels that may block vision, the proportion of tunnels is low. Additionally, although the *lake* map exhibits diverse background colors, the diversity primarily arises from vegetation-covered areas, which occupy a small proportion of the map, resulting in low challenges for the agent. In contrast, single-stage reinforcement learning methods without the CBT strategy fail to converge on the *citystreet*, *downtown*, and *farmland* maps. This suggests that as the visual complexity of scenes and the density of elements increase, directly applying single-stage reinforcement learning is highly challenging and unlikely to converge. These results demonstrate the effectiveness of the CBT strategy.

Robustness under wind gusts and precipitation. In the wind gust simulation experiments, we apply wind velocities in the range of $[2.5, 7.5]m/s$ for forward and lateral directions, and angular rate disturbances of $[0.05, 0.15]rad/s$ around the yaw axis to mimic turbulence and gusts.

E.4 Experiments in Real-world Scenarios

We selected eight video sequences each from the VOT [30], DTB70 [35], and UAVDT [20] datasets to evaluate the transferability of GC-VAT. Specifically, from the VOT benchmark, we chose the videos *car1*, *car3*, *car6*, *car8*, *carchase*, *car16*, *following*, and *car9*. From the DTB70 benchmark, we selected *Car2*, *Car4*, *Car5*, *RcCar4*, *Car8*, *RaceCar*, *RaceCar1*, and *RcCar3*. From the UAVDT benchmark, we chose *S1603*, *S0201*, *S0101*, *S0306*, *S1201*, *S0303*, *S1301*, and *S1701*. We provide qualitative visualizations for representative video sequences. Specifically, Fig. 14 shows the output actions for a video in VOT [30] named *car6*. Fig. 15 shows the output actions for a video in VOT named *car8*. Fig. 16 shows the output actions for a video in DTB70 [35] named *Car4*. Fig. 17 shows the output actions for a video in VOT named *RaceCar1*.

F Limitation

Although we validate the effectiveness of DAT and GC-VAT using real-world images and simple real-world scenarios, deploying the algorithm in truly open environments remains highly challenging. This is primarily due to the presence of numerous similar interfering objects and the high complexity of real-world conditions, which still exhibit a significant gap compared to simulated environments. We will further enhance the algorithm's adaptability and conduct testing in real open-world environments.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See Section 1 for details.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Appendix F for limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Section 4.3 and Section C.1 for complete theoretical proof.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: See Section 5 and Appendix C.2 for all information needed.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: See anonymous homepage for all the data and code.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Section 5.1 and Appendix E for all the experimental settings and details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: See Section 5 and Appendix E for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Appendix E for details.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All aspects of the paper comply with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Section 6 for potential impacts of our paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: See anonymous homepage for details.

    Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We provide a well-organized documentation in anonymous homepage.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in our paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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