Long-term Embodied Visual Tracking with Lightweight Vision-Language-Action Models

Haowei Sun[1*] **Kaining Chen**[1*] **Xutao Wen**[1*] **Xinze Xie**[1*] **Jinwu Hu**[1 2]
Jiaxi Chen[1] **Mingkui Tan**[1 3†] **Shutao Li**[4†]
[1] South China University of Technology, [2] Pazhou Laboratory,
[3] Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, [4] Hunan University

Abstract

Embodied Visual Tracking (EVT) aims to autonomously control an agent to follow a target in 3D space, which is critical for robotic navigation and human-robot interaction. However, robust long-term EVT in real-world scenarios faces two key bottlenecks: frequent tracking failures caused by occlusions and distractors, and the lack of a long-term benchmark for model training and evaluation. To address these, we propose a systematic solution. First, we develop LT-VLA, a lightweight VLA model featuring a Target Memory module to maintain consistent target identification and an Adaptive Execution module to adjust tracking actions based on observation reliability. Second, we construct LTEVT, a unified long-term EVT benchmark comprising 11 high-fidelity indoor and outdoor environments, enriched with over 5,000 assets and 325 human targets to simulate occlusions and distractors. Along with the benchmark, we release the LTEVT-4500K dataset for large-scale training and a comprehensive evaluation toolkit. Extensive experiments demonstrate that our 0.6B model achieves state-of-the-art performance comparable to 7B-scale models while maintaining superior efficiency. Notably, LT-VLA achieves 62.4% average Success Rate on EVT-Bench and runs at 20 FPS on a Unitree G1 robot, delivering a practical and robust solution for real-world deployment.

1 Introduction

Embodied Visual Tracking (EVT) aims to autonomously control an agent to follow a specific target instance in 3D space based on real-time visual observations [5, 22, 47, 46, 48, 45, 49, 35, 11, 31, 32]. It is widely applied in real-world applications such as robotic navigation and human-robot interaction [20, 36, 14]. Unlike passive visual tracking (PVT) [1, 39, 19, 3, 8, 51, 37, 30], which localizes targets within pre-recorded video frames, EVT requires the agent to actively control cameras to maintain the target within the view. Since PVT often falls short in complex and dynamic real-world scenarios, EVT offers a more practical yet more challenging paradigm for open-world tracking applications.

Existing EVT approaches are broadly categorized into pipeline methods and end-to-end methods. **Pipeline methods** [5, 23, 45, 32] decouple the task into separate visual perception and control modules. While intuitive, they suffer from cascading errors between these separated modules. To overcome this, **Reinforcement Learning (RL)-based methods** [22, 46, 49, 11, 31] learn direct pixel-to-action policies. However, their reliance on simulation [28, 10, 11, 31] causes sim-to-real gaps during real-world deployment. Recently, **Vision Language Action (VLA)-based methods** [35, 18] leverage the strong generalization of pretrained models and fine-tune on massive tracking data,

*Equal contribution. Email: sunhoward1105@gmail.com, kainingchen138@gmail.com, wenxutao065@gmail.com, xinzexie@gmx.com
†Corresponding author. Email: mingkuitan@scut.edu.cn, shutao_li@hnu.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

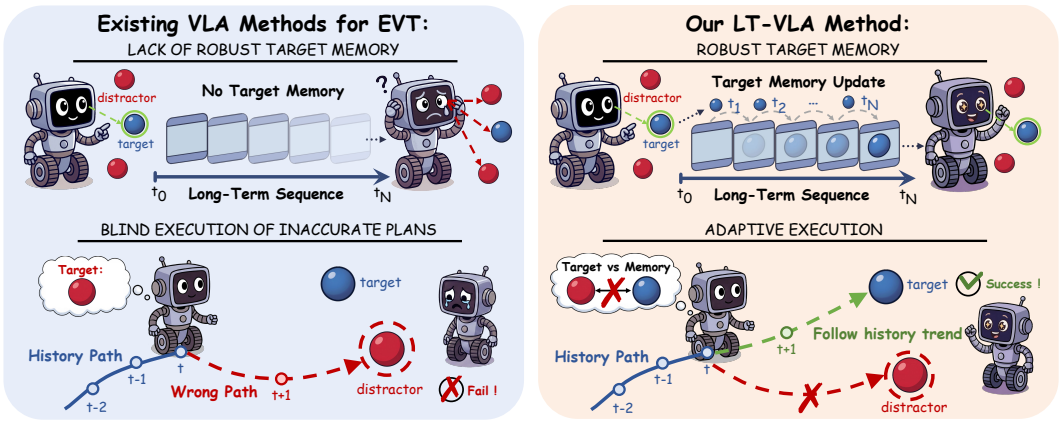


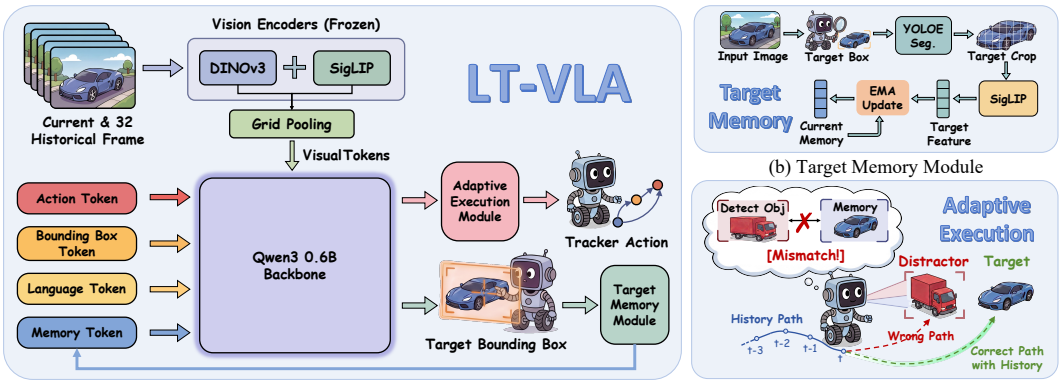
Figure 1: Comparison of existing VLA methods and LT-VLA. We integrate a Target Memory module to maintain consistent target identification against distractors. During execution, we employ an Adaptive Execution module that uses historical paths to adjust actions under noisy observations.

enabling zero-shot transfer to unseen environments. Therefore, VLA-based methods simplify the system design and increase the generalization capability across diverse real-world scenes.

In real-world scenarios, EVT is often not limited to short-horizon target following, but requires the agent to maintain target tracking over extended periods under dynamic visual changes. *Unfortunately*, achieving robust long-term EVT with VLA models remains challenging, partly for the following reasons (Fig. 1). **1) Vast environments with complex interference.** During long-term tracking, the target is often obscured for periods of time or confused with visually similar distractors. This complex interference easily causes existing models to lose the target or track the wrong object due to their lack of robust target memory. Furthermore, noisy observations often cause models to plan incorrect actions. However, existing methods directly execute these inaccurate plans without adaptive adjustment, leading to permanent target loss. **2) Missing unified, long-term EVT benchmark.** Existing EVT benchmarks are typically short-term, restricted to indoor scenes, or lack dense occlusions and distractors in real-world scenarios. This data scarcity restricts both the training and evaluation of long-term EVT models in complex environments.

To address the above limitations, we **first** propose a lightweight VLA model for long-term EVT (called **LT-VLA**). Specifically, we design a Target Memory module that maintains and online-updates the target features. This helps the agent consistently lock onto the correct target under complex interference. Besides, we introduce an Adaptive Execution module that adaptively adjusts tracking actions based on observation reliability, mitigating incorrect planning caused by noisy visual inputs. **Second**, we construct a unified long-term EVT benchmark, named **LTEVT**. Specifically, we provide 5 high-fidelity indoor and 6 outdoor environments across two different simulators, featuring dense distractor interference and frequent occlusions. Furthermore, we release the **LTEVT-4500K** dataset for large-scale training along with a toolkit for comprehensive long-term EVT evaluation. Our contributions are summarized as follows:

- **A Lightweight VLA Model for Long-term EVT.** We propose a VLA-based tracker featuring a target memory module for accurate target identification against distractors and an adaptive execution module for action adjustment under noisy observations.
- **A Unified Long-term EVT Benchmark.** We construct a unified benchmark comprising 11 high-fidelity environments with dense distractors and occlusions. Along with the benchmark, we release a large-scale dataset and a toolkit for both model training and evaluation.
- **Extensive Validation.** Extensive experiments demonstrate our state-of-the-art performance and real-time efficiency. Notably, our 0.6B model achieves performance comparable to 7B-scale models and runs at 20 FPS on a Unitree G1 robot, highlighting its superior efficiency.



(a) Overview of LT-VLA

(b) Target Memory Module

(c) Adaptive Execution Module

Figure 2: Overview of Long-Term VLA (LT-VLA). (a) LT-VLA follows a Vision-Language-Action paradigm to map visual observations and language instructions directly into tracking actions. (b) The Target Memory module maintains and online-updates target features to ensure consistent identification against complex distractors. (c) The Adaptive Execution module leverages historical trajectories to adjust tracking actions under noisy observations, preventing permanent target loss.

2 Embodied Visual Tracking: Problem Definition

Embodied Visual Tracking (EVT) aims to autonomously control a mobile agent to follow a specific target. This task poses two key challenges: maintaining an accurate memory of the target, and executing robust actions under distractors and occlusions. We hereby formalize EVT as a vision-language control problem, defined by the observation space, action space, and success criterion described below, and solved by an EVT tracker. Furthermore, we discuss the necessity and challenges of long-term EVT in real-world applications.

Observation Spaces $\mathcal{O}$. At each time step T, the agent receives a sequence of egocentric RGB images $O_T = \{o_{T-K+1}, \ldots, o_{T-1}, o_T\}$ over horizon K. Besides, a language instruction I is provided at the beginning of the episode to specify the target identity.

Action Spaces $\mathcal{A}$. The EVT tracker outputs n future waypoints $W_T = \{w_1, w_2, \ldots, w_n\}$. Each waypoint $w_k \in \mathbb{R}^3$ represents the tracker's desired 3D position (x, y, z) at time step k.

EVT Tracker. We define an EVT tracker as an embodied agent parameterized by a policy π_θ, which maps historical visual observations and the target instruction to a future waypoint sequence, i.e.,

$$W_T = \pi_\theta(O_T, I). \tag{1}$$

Success criterion. A successful EVT episode requires the agent to keep the target centered in the view under challenging conditions such as occlusions and distractors. Metrics are detailed in Sec. 5.1.

Challenges of long-term EVT. Real-world applications like autonomous navigation require robots to maintain target tracking over extended horizons. However, long-term EVT is extremely challenging primarily because complex distractors often mislead target identification, while dynamic visual changes cause unstable trajectory planning, leading to permanent tracking failure.

3 Long-term EVT with Vision Language Action Model

Long-term embodied visual tracking in the real world is extremely challenging, primarily because frequent occlusions and visually similar distractors easily lead to erroneous action plans. To address these problems, we propose Long-Term VLA (LT-VLA), an efficient VLA model for embodied tracking under complex interference. As shown in Fig. 2, LT-VLA introduces a robust target memory to ensure accurate target identification against distractors and occlusions, and an adaptive execution module to adjust incorrect action plans caused by noisy observations.

LT-VLA Overview. Specifically, as shown in Fig. 2(a), LT-VLA is built upon a lightweight Qwen3-0.6B [33] backbone, using frozen DINOv3 [29] and SigLIP [42] as visual encoders. Following [35], we apply a grid pooling strategy to the extracted features of the current and 32 historical frames to generate visual tokens. Moreover, to adapt the LLM for embodied visual tracking, we introduce three

Algorithm 1 The pipeline of proposed LT-VLA

Require: Current image $\mathcal{I}_t$, Language prompt I, Historical frames O_{t-1}, Target memory $\mathcal{M}_{t-1}$, Historical velocity window $\mathcal{H}_v$, LT-VLA model $\texttt{LT-VLA}(\cdot)$, Feature extractor $\texttt{Desc}(\cdot)$, Segmentation model $\texttt{Seg}(\cdot)$
Ensure: Executed action v_t, Updated target memory $\mathcal{M}_t$

Predict waypoints, box and confidence via LT-VLA: $[\hat{W}_t, B_t, c_t] \leftarrow \texttt{LT-VLA}(\mathcal{I}_t, I, O_{t-1}, \mathcal{M}_{t-1})$
Crop target patch $\mathcal{I}_{\text{tar}}$ from $\mathcal{I}_t$ using B_t, and predict target mask $M_{\text{tar}} \leftarrow \texttt{Seg}(\mathcal{I}_{\text{tar}})$
Obtain masked patch $\tilde{\mathcal{I}}_{\text{tar}} \leftarrow M_{\text{tar}} \odot \mathcal{I}_{\text{tar}}$, and extract target instance feature $f_t \leftarrow \texttt{Desc}(\tilde{\mathcal{I}}_{\text{tar}})$
Compute similarity $\mathcal{S}_t$ for reliability assessment via Eq. (3)
if $c_t \geq c_0$ **and** $\mathcal{S}_t \geq s_0$ **then**
    // State: Reliable
    Calculate velocity v_t based on planned waypoint $\hat{W}_t[0]$
    Update target memory via EMA: $\mathcal{M}_t \leftarrow \gamma \mathcal{M}_{t-1} + (1-\gamma) f_t$
else if $c_t < c_0$ **then**
    // State: Lost
    Calculate velocity v_t to recover the target with predicted waypoint $\hat{W}_t[0]$
    Keep previous target memory: $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1}$
else
    // State: Distracted
    Adjust action via averaged historical velocity: $v_t \leftarrow \frac{1}{K} \sum \mathcal{H}_v$
    Keep previous target memory: $\mathcal{M}_t \leftarrow \mathcal{M}_{t-1}$
end if
Update velocity window: $\mathcal{H}_v \leftarrow \mathcal{H}_v \cup \{v_t\}$
return $v_t, \mathcal{M}_t$

learnable tokens alongside standard visual and language inputs: a *memory token* to inject the target's visual identity, a *bounding box token* to enable spatial grounding, and an *action token* to plan future waypoints. Notably, the Target Memory module (detailed in Sec. 3.1) uses the predicted bounding box to construct and online-update the target memory, which then forms the *memory token* (Fig. 2(b)). During execution, we decode the hidden states of bounding box and action tokens with lightweight heads to obtain the target boxes and planned waypoints. Our Adaptive Execution module (detailed in Sec. 3.2) then adjusts the output actions based on the observation reliability to control the tracker (Fig. 2(c)). The pseudocode of LT-VLA is shown in Algorithm 1.

3.1 Robust Target Memory for Long-Term Tracking

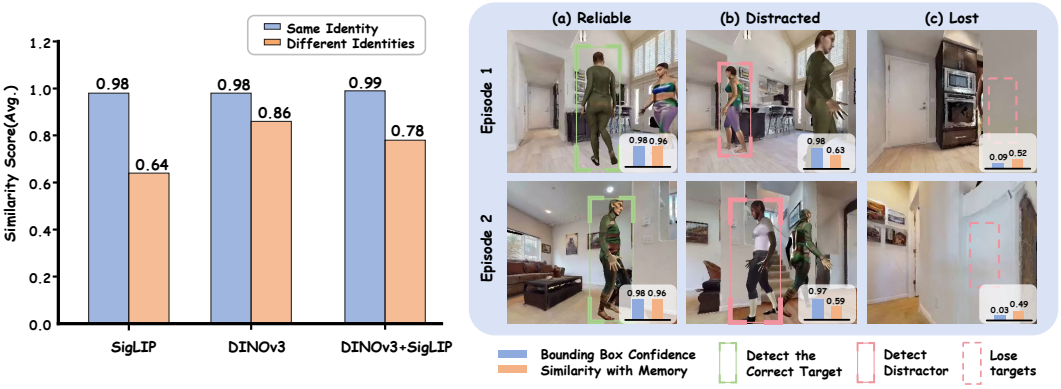
We seek to construct a target memory that can reliably distinguish the tracking target from other visually similar instances of the same category, while remaining robust to viewpoint shifts and environmental variations. Since LT-VLA uses DINOv3 and SigLIP as visual encoders, we aim to directly leverage their feature representations to build this instance-specific memory.

Empirical Choice of Feature Extractor. To select the most suitable encoder, we evaluate the instance-level discriminative power of DINOv3, SigLIP, and their concatenated features. For each object instance, we sample 100 images from different viewpoints, extract their single-view features, and aggregate them via average pooling. As shown in Fig. 3(a), SigLIP features exhibit the highest capability to distinguish different instances of the same class (a similarity gap of 0.34 between identical and distinct instances). Consequently, we select SigLIP as our feature extractor $\texttt{Desc}(\cdot)$.

Memory Construction and Online Update. To accurately localize the target, LT-VLA incorporates a bounding box head to predict the target bounding box B_t and a confidence score c_t for each incoming image $\mathcal{I}_t$. When the confidence exceeds a predefined threshold ($c_t > c_0$), we use B_t to crop the target patch $\mathcal{I}_{\text{tar}}$. To further suppress background noise, we employ a segmentation model $\texttt{Seg}(\cdot)$ (we use YOLOE [34] for its speed-performance balance) to extract precise target mask $M_{\text{tar}} = \texttt{Seg}(\mathcal{I}_{\text{tar}})$. The mask is then applied to the target patch: $\tilde{\mathcal{I}}_{\text{tar}} = M_{\text{tar}} \odot \mathcal{I}_{\text{tar}}$. Subsequently, we obtain the target feature with extractor: $f_t = \texttt{Desc}(\tilde{\mathcal{I}}_{\text{tar}})$. During execution, the target memory $\mathcal{M}$ is initialized with a learned default token. When current visual inputs are reliable (detailed in Sec. 3.2), we dynamically update $\mathcal{M}$ with the target feature via an exponential moving average:

$$\mathcal{M}_t = \gamma \mathcal{M}_{t-1} + (1-\gamma) f_t, \tag{2}$$

where γ is the momentum factor. This updated memory $\mathcal{M}_t$ is then fed back into LT-VLA as the new memory token for the next step, prompting the tracker with correct target identity. Additionally, if a



(a) Comparison of feature similarities between the same identity and different identities.

(b) Comparison between reliable observation, tracking a distractor, and target loss.

Figure 3: Motivation for Target Memory and Adaptive Execution module. (a) SigLIP provides superior instance-level discriminability for robust target identification. (b) Jointly using box confidence and feature similarity effectively identifies target loss and distractor interference.

target reference image is provided, we can directly extract its feature using $\texttt{Desc}(\cdot)$ to initialize $\mathcal{M}$. This explicit visual prior further enhances the accuracy of target localization in the first frame.

3.2 Adaptive Execution Module based on Observation Reliability

Even with a robust target memory, environmental disturbances like camera jitter from robot motion, harsh illumination, and obstacle occlusions can still corrupt visual inputs and lead to erroneous action plans. To systematically resolve this, we introduce an adaptive execution module. It evaluates the reliability of the current observation using the target memory (in Sec. 3.1) and dynamically adjusts the motion planning, enabling the tracker to handle environmental interference.

Observation Reliability Assessment. Intuitively, a reliable observation requires the target to be both visible and correctly identified. However, in long-horizon tracking, occlusions and visually similar distractors often compromise these conditions, leading to unreliable observations. We illustrate the impact of these types of interference on LT-VLA in Fig. 3. As shown in Fig. 3(b), bounding box confidence drops significantly ($0.98 \rightarrow 0.09$) when the target is occluded. Conversely, when the model detects a wrong target, the box confidence remains high, but the feature similarity with the target memory is low ($0.96 \rightarrow 0.49$). Therefore, we jointly use the bounding box confidence c_t and the cosine similarity $\mathcal{S}_t$ to measure observation reliability, where $\mathcal{S}_t$ is given by:

$$\mathcal{S}_t = \frac{f_t \cdot \mathcal{M}_{t-1}}{\|f_t\| \cdot \|\mathcal{M}_{t-1}\|}. \tag{3}$$

Specifically, the observation is considered *Reliable* only if c_t and $\mathcal{S}_t$ simultaneously exceed their respective thresholds c_0 and s_0. If $c_t < c_0$, the target is considered *Lost* due to severe occlusion or moving out of frame. Conversely, if $c_t \geq c_0$ but $\mathcal{S}_t < s_0$, the state is identified as *Distracted*, indicating the model may mistakenly ground a visually similar distractor. These state signals are then used by LT-VLA to actively adjust its trajectory planning and action execution.

Reliability-Aware Adaptive Execution. After LT-VLA outputs the planned waypoints $\hat{W}_t$, the bounding box confidence c_t, and the updated memory $\mathcal{M}_t$, we dynamically adjust the actions based on the observation reliability. To guide the tracker during tracking failures, we maintain a historical velocity window $\mathcal{H}_v = \{v_{t-K}, \ldots, v_{t-1}\}$ of horizon K. For a *Reliable* observation, we directly execute the first waypoint in $\hat{W}_t$ and update the velocity into $\mathcal{H}_v$. When the target is *Lost*, we follow the planned trajectories to recover the occluded target. Additionally, when the tracker is *Distracted*, following the current plan will lead to failure. Therefore, we use $\mathcal{H}_v$ to derive a reliable command v_t^*:

$$v_t^* = \frac{1}{K} \sum_{i=1}^{K} v_{t-i}. \tag{4}$$

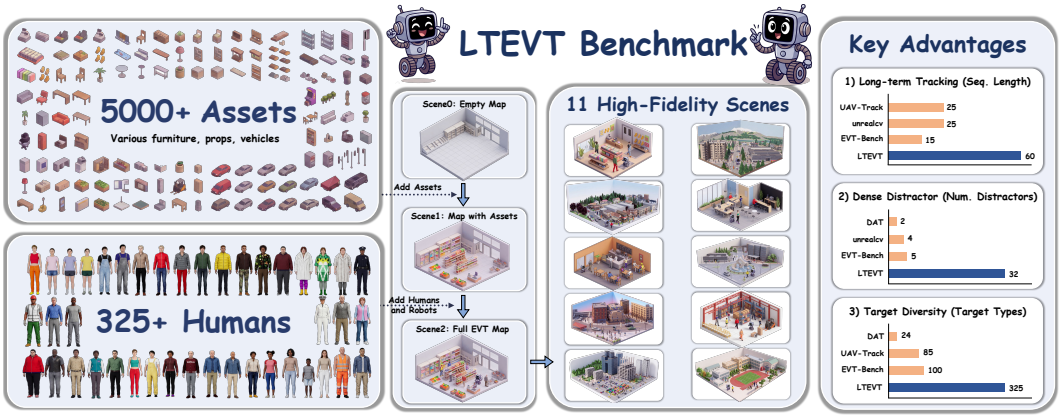


Figure 4: Overview of LTEVT benchmark. Our dataset features 11 high-fidelity indoor and outdoor scenarios with 325 unique human targets and over 5,000 diverse assets, covering long-horizon trajectories with complex interference.

Table 1: Comparison between LTEVT benchmark and existing tracking datasets. The "–" symbol indicates that the original paper does not provide the episode length in seconds.

Dataset	In/Outdoor	Language	Target Types	Avg. Length (s)	Size (K)
DAT [31]	✗/✓	✗	24	3 s	✗
D-VAT [11]	✓/✓	✗	1	–	✗
IndoorUAV [21]	✓/✗	✓	0	–	768
UAV-Track [44]	✗/✓	✓	85	25 s	890
unrealcv [28]	✓/✗	✗	4	25 s	✗
EVT-bench [35]	✓/✗	✓	100	15 s	855
LTEVT (Ours)	✓/✓	✓	**325**	**60 s**	**4500**

Training Objective. We train LT-VLA with imitation learning. Specifically, we optimize the model's spatial grounding and motion planning capabilities with the following loss $\mathcal{L}$:

$$\mathcal{L} = \lambda_{\text{box}} \cdot \mathcal{L}_{\text{box}} + \lambda_{\text{act}} \cdot \mathcal{L}_{\text{act}}, \tag{5}$$

where $\mathcal{L}_{\text{box}}$ comprises an L_1 loss for bounding box coordinates and a cross-entropy loss for confidence. $\mathcal{L}_{\text{act}}$ is the MSE loss for the planned waypoints, and $\lambda_{\text{box}}, \lambda_{\text{act}}$ are scaling hyperparameters.

4 LTEVT: A Long-term EVT Benchmark with Dense Interference

We aim to model two of the most challenging aspects of EVT tasks: long-term sequences and dense interference. To this end, we construct LTEVT, a unified benchmark across two simulation platforms with 5 high-fidelity indoor and 6 outdoor environments, featuring dense distractors and long-horizon tracking trajectories. Furthermore, we release **LTEVT-4500K**, a large-scale dataset to train VLA models for embodied visual tracking. As shown in Tab. 1, compared to existing benchmarks, LTEVT features significantly longer episodes, more diverse target types, and a much larger training dataset.

4.1 Benchmark Construction

Diverse Scenarios. To ensure realistic rendering and physical interactions, LTEVT is built upon two state-of-the-art platforms: Isaac Sim [25] and CARLA [12]. As shown in Fig. 4, we construct 5 indoor environments in Isaac Sim (e.g., markets, museums, warehouse, and offices) and 6 diverse outdoor maps in CARLA (e.g., urban cities, mountainous terrains, and villages). These environments cover most open-world EVT scenarios. Furthermore, we place over 5000 distinct 3D assets in the scenes to simulate realistic visual occlusions and evaluate the agent's obstacle avoidance capabilities.

Various Targets and Dense Distractors. To improve the model's robustness against appearance variations, our benchmark features 325 distinct human target assets (Fig. 4). To simulate realistic human locomotion, we randomize the target speed between 0.0 and 2.0 m/s. In addition, we place

dense moving distractors in the scenes (32 in indoor scenes and 16 in outdoor scenes). This poses significant challenges for both precise target memory and collision avoidance.

Tracking instructions. We leverage Qwen3-VL [2] to generate 5 unique appearance descriptions for each human target. These descriptions are then inserted into instruction templates to form complete commands. This approach ensures linguistic diversity, forcing the agent to identify the target based on varied descriptive cues. *See Appendix for detailed instruction templates.*

Task Settings. Following [35], we define three tasks in LTEVT. **Single Target Tracking (STT)** involves following a single target without any interference. **Distracted Tracking (DT)** evaluates tracking robustness against moving distractors. **Ambiguity Tracking (AT)** evaluates the agent's ability to identify the correct target among visually identical distractors using spatial understanding. We implement all three tasks in outdoor maps, while only the DT task is configured for indoor scenes.

4.2 LTEVT-4500K: Large-Scale Dataset for VLA Training

To train LT-VLA for robust embodied tracking with imitation learning, we construct a large-scale dataset consisting of 4,500K tracking sequences generated from our proposed LTEVT and 800K visual grounding samples. The tracking data provides essential state-action pairs for motion planning in complex environments, while the grounding data strengthens the model's ability to identify specific individuals under textual guidance and distinguish targets from similar distractors.

Expert Action Generation. To provide high-quality supervision for imitation learning, we develop a hybrid expert policy to generate the ground-truth action labels $\mathbf{a}_t = [v_x, v_y, \omega_{yaw}]$. When the target is visible, the expert uses a PID [24] controller guided by the target bounding box. The controller adjusts the robot heading to move the bounding box back to the image center and regulates the forward speed based on the box area to maintain a consistent tracking distance.

When the target is occluded, the bounding box disappears. To handle these cases, the expert switches to an A^* path planner [16] using the target's ground-truth 3D positions. The planner generates a collision-free path to the target. Subsequently, the robot follows the generated waypoints with a PID controller, allowing it to safely navigate around obstacles. We then integrate these velocity commands to obtain n future waypoints W_T, as required by the action space in Sec. 2.

Tracking Data. Using the hybrid expert policy, we collect 4,500K samples from LTEVT. To cover household environments, we further sample 1,200K samples from EVT-Bench [35]. For all tracking sequences, ground-truth bounding boxes are derived from instance segmentation masks provided by the simulators. This extensive data collection provides a high-fidelity dataset that enables LT-VLA to learn robust control laws across varied physical settings.

Visual Grounding Data. To improve target identification under textual guidance, we incorporate 800K visual grounding samples into our training recipe. We first use YOLOE [34] to segment high-quality human figures from SYNTH-PEDES [52]. We then randomize the scale of these figures and paste them onto various locations within background images from the Places365 [50] dataset.

5 Experiments

5.1 Experimental Settings

Experimental Setup. We evaluate the performance of all methods on EVT-Bench [35] and the proposed LTEVT. Specifically, we train separate models using datasets from EVT-Bench, LTEVT Indoor, and LTEVT Outdoor environments, then evaluate each model within its corresponding test scenes.

Metrics. We report three metrics: *Success Rate (SR)* calculates the percentage of successfully completed episodes, *Tracking Rate (TR)* measures the proportion of steps where the target is correctly tracked, and *Collision Rate (CR)* records the frequency of agent collisions during the tracking.

Baselines. We compare LT-VLA with 9 baselines, grouped into 3 categories: the RL-based tracker EVT [49], pipeline-based EVT trackers including IBVS [15], OR-Track [38], PoliFormer [41], Follow-Anything [23] and OA-VAT [32], and VLA-based methods such as Uni-NaVid [43], OpenTrackVLA [18], and TrackVLA [35]. *Related works are detailed in the Appendix.*

Table 2: Results on EVT-bench. TrackVLA (7B) is treated as the oracle performance. * denotes OpenTrackVLA trained with the same data as ours, whereas the original version uses a multi-task data mixture. **Bold** represents the best while underline represents the second.

Method	STT Task			DT Task			AT Task			Average		
	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓
TrackVLA (7B) [35]	85.1	78.6	1.6	57.6	63.2	5.8	50.2	63.7	17.1	64.3	68.5	8.2
IBVS [15]	42.9	56.2	3.8	10.6	28.4	6.1	15.2	39.5	4.9	22.9	41.4	4.9
ORTrack [38]	20.9	39.4	12.6	5.3	31.7	15.8	12.5	43.5	9.6	12.9	38.2	12.7
PoliFormer [41]	4.7	15.5	40.1	2.6	13.2	44.5	3.0	15.4	41.5	3.4	14.7	42.0
EVT [49]	24.4	39.1	42.5	3.2	11.2	47.9	17.4	21.1	45.6	15.0	23.8	45.3
Uni-NaVid [43]	25.7	39.5	41.9	11.3	27.4	43.5	8.3	28.6	43.7	15.1	31.8	43.0
FollowAnything [23]	38.7	67.6	35.7	13.8	49.6	35.6	18.0	56.1	33.1	23.5	57.8	34.8
OA-VAT [32]	40.3	41.6	7.9	14.1	33.8	21.9	18.9	40.1	27.8	24.4	38.5	19.2
OpenTrackVLA [18]	75.8	78.4	9.1	39.9	57.4	15.2	**55.5**	70.9	14.4	57.1	68.9	12.9
OpenTrackVLA* [18]	82.9	85.8	5.8	31.4	57.4	16.4	37.8	64.8	8.8	50.7	69.3	10.3
LT-VLA (Ours, 0.6B)	**87.8**	**86.1**	**3.2**	**51.5**	**65.8**	**5.1**	47.8	**72.6**	**4.3**	**62.4**	**74.8**	**4.2**

Table 3: Results on LTEVT. **Bold** represents the best while underline represents the second.

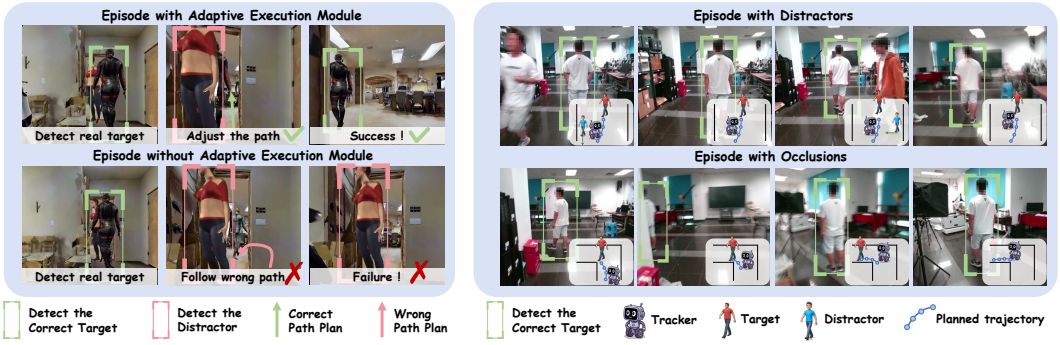
Method	Outdoor STT		Outdoor DT		Outdoor AT		Indoor DT			Avg. SR
	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	SR ↑	TR ↑	CR ↓	
OR-Track [38]	1.9	30.8	4.4	40.8	0.8	32.5	1.4	8.2	1.7	2.1
EVT [49]	0.5	22.4	0.3	23.4	0.2	22.3	1.7	8.4	3.8	0.7
FAn [23]	8.3	48.5	9.8	42.1	5.3	41.7	5.6	7.5	2.3	7.3
OA-VAT [32]	8.5	38.7	4.4	42.6	0.8	37.0	0.2	8.7	1.9	3.5
OpenTrackVLA [18]	45.1	58.1	18.2	52.1	16.3	49.1	8.2	15.5	3.5	22.0
LT-VLA (Ours)	**48.9**	**61.3**	**24.6**	**55.0**	**20.1**	**54.2**	**12.3**	**18.2**	**1.6**	**26.5**

Implementation Details. We train specialized models for LTEVT Outdoor, LTEVT Indoor, and EVT-Bench, respectively. Each model is trained on a combination of its corresponding tracking data (including all STT, DT, and AT tasks) and visual grounding samples (detailed in Sec. 4.2). Specifically, we employ different token configurations for different data. For visual grounding data, we only activate image, language, and bounding box tokens. For tracking data, we retain all input tokens for action planning. To enhance tracking performance without reliable target memory, we drop out memory tokens with a 10% probability and replace them with a learnable initialization token. We train LT-VLA on 8 NVIDIA A800 (80GB) GPUs for 30 hours. *See Appendix for more details.*

5.2 Comparison Experiments

Effectiveness and Efficiency on EVT-bench. We treat TrackVLA (7B) [35] as the oracle model and aim to bridge the performance gap using our 0.6B LT-VLA. As shown in Tab. 2, despite using less than 10% of the oracle's parameters, LT-VLA achieves a higher average TR (74.8% vs. 68.5%) and a halved average CR (4.2% vs. 8.2%), exhibiting a marginal performance gap of only 3.0% in average SR (62.4% vs. 64.3%). Furthermore, when trained on identical datasets, LT-VLA outperforms the state-of-the-art OpenTrackVLA [18] by 23.1% in average SR. These results validate the robust tracking, high efficiency, and superior safety of LT-VLA for real-world robot deployment.

Effectiveness on LTEVT. As shown in Tab. 3, LT-VLA consistently outperforms existing EVT models across all test scenarios. Our model achieves a 20.5% improvement in average Success Rate (SR) over OpenTrackVLA [18] method. Furthermore, LT-VLA maintains a collision rate of 1.6% in indoor environments, validating its precise tracking and navigation planning capabilities. Notably, LT-VLA achieves the greatest gains in the Outdoor DT and Indoor DT tasks, with gains of 35.2% ($18.2\% \rightarrow 24.6\%$) and 50.0% ($8.2\% \rightarrow 12.3\%$) in average SR over the SOTA method, respectively. This highlights LT-VLA's superior robustness against dense interference in long-term tracking.



(a) Comparison between the proposed Adaptive Execution module and direct execution.

(b) Real-world deployment on Unitree G1 robot. LT-VLA follows the target under complex distractors and occlusions.

Figure 5: Qualitative validation of LT-VLA. (a) The proposed Adaptive Execution module corrects the tracking action when the model identifies a distractor. (b) LT-VLA robustly follows the correct target on the Unitree G1 robot despite real-world distractors and occlusions.

Table 4: Ablation results on EVT-Bench. **Bold** represents the best while underline denotes the second.

Variant	STT			DT			AT			Avg SR
	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	SR ↑	TR ↑	CR ↓	
w/o Target Memory	85.2	83.1	4.6	36.8	58.4	12.6	38.4	66.2	12.2	53.5
w/o Adaptive Execution	87.1	84.0	4.2	45.9	63.2	8.9	45.8	68.5	11.7	59.6
LT-VLA (Ours)	**87.8**	**86.1**	**3.2**	**51.5**	**65.8**	**5.1**	**47.8**	**72.6**	**4.3**	**62.4**

5.3 Ablation Experiments

We conduct ablation studies on EVT-Bench to verify the necessity of our proposed Target Memory module and Adaptive Execution module.

Effectiveness of Target Memory Module. As shown in Tab. 4 (Line 1), LT-VLA outperforms the variant without the target memory by an average of 16.6% ($53.5\% \rightarrow 62.4\%$) in Success Rate across three settings. Notably, in the DT task, which poses the greatest challenge for target identification, our model achieves a 39.9% ($36.8\% \rightarrow 51.5\%$) improvement. This gain demonstrates that the proposed Target Memory module effectively maintains target information over long sequences.

Effectiveness of Adaptive Execution Module. We compare LT-VLA with a version without the Adaptive Execution module. As shown in Tab. 4 (Line 2), our model achieves an average improvement of 4.7% ($59.6\% \rightarrow 62.4\%$) in average Success Rate on EVT-Bench. We further provide qualitative visualizations in Fig. 5(a). As shown in Fig. 5(a) (Line 1), when the similarity between the detected target and target memory is low, this module corrects the action to track the correct target. Without this execution module, the model follows erroneous trajectory plans and fails (Fig. 5(a) (Line 2)).

5.4 Experiments in Real Robots

Effectiveness and efficiency on Unitree G1 Humanoid Robot. We evaluate the real-world effectiveness and efficiency of LT-VLA through zero-shot deployment on a Unitree G1 humanoid robot. RGB streams at 384×384 resolution are captured by the onboard camera and transmitted via Wi-Fi to a remote NVIDIA RTX A6000 server for inference. Our LT-VLA model then processes these visual inputs to predict velocity commands, which are sent back to the robot's onboard PC. These commands are then processed by the `unitree_sdk2_python` interface and executed by an onboard walking controller for movement. This setup enables real-time, closed-loop control during the evaluation.

We evaluate LT-VLA in challenging scenes with dynamic distractors and complete occlusions. As illustrated in Fig. 5(b), LT-VLA locks on the correct person even when distractors pass through the field of view. When the target is occluded, the model plans a recovery path and resumes tracking immediately. Furthermore, the model achieves an inference speed of 20 FPS, fulfilling the real-time control requirements for humanoid robotic tasks.

6 Conclusion and Potential Impacts

In this paper, we propose LT-VLA, a lightweight VLA model designed for robust long-term Embodied Visual Tracking (EVT). Specifically, we introduce a Target Memory module to maintain consistent target identification and an Adaptive Execution module to adjust tracking actions under noisy observations. We also construct LTEVT, a unified long-term EVT benchmark featuring 11 high-fidelity indoor and outdoor environments with dense distractors and frequent occlusions. Along with the benchmark, we release the LTEVT-4500K dataset and a comprehensive evaluation toolkit. Extensive experiments demonstrate that our 0.6B model achieves state-of-the-art performance comparable to 7B-scale models while maintaining superior efficiency for real-world deployment.

The LTEVT benchmark and LT-VLA method have the potential to impact several research areas, including: 1) human-robot interaction and navigation, 2) planning and obstacle avoidance in complex scenarios, and 3) scene understanding in open-world environments. By providing both an efficient architecture and a challenging benchmark, this work paves the way for more practical and reliable embodied visual tracking in open-world applications.

Acknowledgements

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

References

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

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

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

[4] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

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

[6] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning*, 2025.

[7] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[8] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14572–14581, June 2023.

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

[10] 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.

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

[12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[13] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *arXiv preprint arXiv:2505.23705*, 2025.

[14] Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Eric Wang. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3043–3061, Toronto, Canada, July 2023. Association for Computational Linguistics.

[15] Meenakshi Gupta, Swagat Kumar, Laxmidhar Behera, and Venkatesh K Subramanian. A novel vision-based tracking algorithm for a human-following mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1415–1427, 2016.

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

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

[18] Heting Ying Kyusong Lee and Tiancheng Zhao. Opentrackvla: Open-source visual language action model for visual navigation and following. https://github.com/om-ai-lab/OpenTrackVLA, 2025.

[19] 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.

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

[21] Xu Liu, Yu Liu, Hanshuo Qiu, Yang Qirong, and Zhouhui Lian. Indooruav: Benchmarking vision-language uav navigation in continuous indoor environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23864–23872, 2026.

[22] 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.

[23] 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.

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

[25] NVIDIA. Isaac Sim.

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

[27] Zhangyang Qi, Zhixiong Zhang, Yizhou Yu, Jiaqi Wang, and Hengshuang Zhao. Vln-r1: Vision-language navigation via reinforcement fine-tuning. *arXiv preprint arXiv:2506.17221*, 2025.

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

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

[30] 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.

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

[32] Haowei Sun, Kai Zhou, Hao Gao, Shiteng Zhang, Jinwu Hu, Xutao Wen, Qixiang Ye, and Mingkui Tan. Instance-level visual active tracking with occlusion-aware planning. *arXiv preprint arXiv:2604.21453*, 2026.

[33] Qwen Team. Qwen3 technical report, 2025.

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

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

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

[37] 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.

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

[39] 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.

[40] Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. Correctnav: Self-correction flywheel empowers vision-language-action navigation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18737–18745, 2026.

[41] Kuo-Hao Zeng, Zichen Zhang, Kiana Ehsani, Rose Hendrix, Jordi Salvador, Alvaro Herrasti, Ross Girshick, Aniruddha Kembhavi, and Luca Weihs. Poliformer: Scaling on-policy rl with transformers results in masterful navigators. In *Conference on Robot Learning*, pages 408–432. PMLR, 2025.

[42] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[43] Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024.

[44] Qiyao Zhang, Shuhua Zheng, Jianli Sun, Chengxiang Li, Xianke Wu, Zihan Song, Zhiyong Cui, Yisheng Lv, and Yonglin Tian. Uav-track vla: Embodied aerial tracking via vision-language-action models. *arXiv preprint arXiv:2604.02241*, 2026.

[45] Fangwei Zhong, Xiao Bi, Yudi Zhang, Wei Zhang, and Yizhou Wang. Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3705–3714, 2023.

[46] 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.

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

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

[49] 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.

[50] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[51] 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.

[52] Jialong Zuo, Jiahao Hong, Feng Zhang, Changqian Yu, Hanyu Zhou, Changxin Gao, Nong Sang, and Jingdong Wang. Plip: Language-image pre-training for person representation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Supplementary Materials for "Long-term Embodied Visual Tracking with Lightweight Vision-Language-Action Models"

Contents

A Related Work **14**

A.1 Embodied Visual Tracking 14

A.2 Vision-Language-Action models for Robotics 14

B More Details of LTEVT Benchmark **15**

B.1 More Details of Indoor Maps Construction in Isaac Sim 15

B.2 More Details of Outdoor Maps Construction in Carla 16

B.3 More Details of Instruction Templates 16

C Implementation Details and Extended Analysis **17**

C.1 Hyperparameter Details and Sensitivity Analysis 17

C.2 Statistical Variance and Stability Analysis 17

D Limitations **18**

A Related Work

A.1 Embodied Visual Tracking

Embodied Visual Tracking (EVT) aims to control a mobile agent to continuously follow a specific target. Existing EVT methods can be broadly categorized into pipeline-based and end-to-end approaches, depending on how they integrate visual perception and motion planning.

Pipeline-based EVT methods. Pipeline approaches [5, 23, 45, 32] decouple EVT into separate perception and control modules. An object tracker first localizes the target, and a downstream controller (e.g., PID [24]) generates motion commands to keep the target centered. For instance, FollowAnything [23] uses visual foundation models (e.g., SAM [17], DINOv2 [26]) to extract features from a reference image for target matching and employs a PID controller for motion execution. OA-VAT [32] combines a DINOv3-based [29] object tracker with a diffusion [9]-based planner to handle target occlusions. While modular design enables clear task separation, it suffers from error accumulation, leading to irreversible tracking failure under occlusion or distractor interference.

End-to-end EVT Methods. End-to-end methods eliminate modular design by directly mapping observations to actions. A typical approach is reinforcement learning (RL), which learns policies from simulated interaction and enables low-latency tracking control. However, RL-based methods [22, 46, 49, 11, 31] rely heavily on simulation for training, which leads to significant sim-to-real gaps during model deployment. In contrast, recent VLA-based approaches [35, 18] train vision-language-action models on large-scale datasets by leveraging pretrained LLM backbones. Among them, TrackVLA [35] trains a unified model combining Vicuna-7B and a diffusion-based planner, achieving strong zero-shot tracking in real environments. Our method follows this end-to-end VLA paradigm but aims to achieve high performance with smaller models, improving efficiency and deployability.

A.2 Vision-Language-Action models for Robotics

Vision-Language-Action (VLA) models map observations and instructions directly to actions by inheriting visual-language knowledge from pretrained large language models and fine-tuning on robot data. This framework has been applied to tasks like manipulation [7, 6, 13], navigation [43, 27, 40], and tracking [35, 18], showing good generalization in real-world settings. For example, π_0 [7] combines the PaliGemma [4] model with a flow-matching action expert, enabling precise

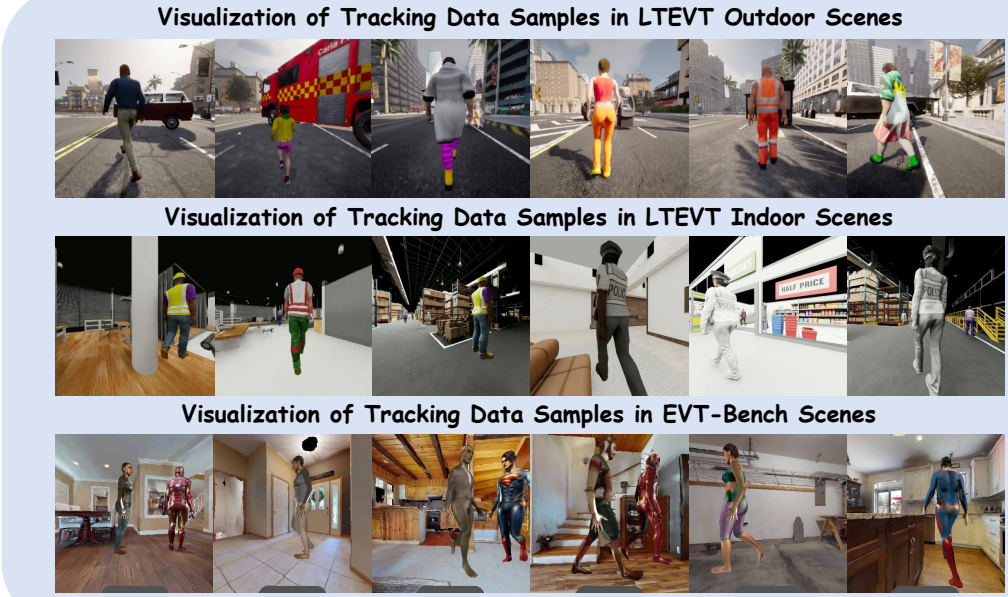


Figure 6: Visualization of diverse training samples from LTEVT benchmark and EVT-Bench.

and continuous control for robotic manipulation. Similarly, Uni-Navid [43] adopts a VLM-based architecture for navigation in dynamic environments. Despite these advances, most VLA models rely on large backbones with heavy computational costs and there is still a lack of lightweight models that achieve strong generalization with high efficiency.

B More Details of LTEVT Benchmark

In this section, we provide comprehensive implementation details of the LTEVT benchmark. Specifically, Appendix B.1 and Appendix B.2 elaborate on the construction processes of indoor maps using Isaac Sim [25] and outdoor environments within Carla, respectively. Furthermore, a visual overview of the LTEVT-4500K dataset across all diverse scenarios is presented in Fig. 6.

B.1 More Details of Indoor Maps Construction in Isaac Sim

We construct diverse simulated environments in Isaac Sim [25] to generate training data and evaluate our method. Specifically, we design five public scenes with distinct spatial layouts, object distributions, and functional semantics, aiming to capture a wide range of public real-world complexities.

The market scene spans 253×256 square meters and is characterized by diverse functional regions. Its dense object distribution and diverse visual patterns introduce significant background clutter, making it suitable for evaluating tracking robustness in complex and crowded environments.

The museum scene spans 23×30 square meters, with walled exhibition rooms. Paintings and structured layouts create partial occlusions, requiring stable tracking under varying visibility.

The school scene spans 100×79 square meters, featuring large open space and structured regions. It tests the ability to handle tracking and transitions between indoor and outdoor environments.

The office scene spans 45×35 square meters, presenting a complex indoor layout. Its narrow passages and frequent occlusions challenge the agent's short-term target re-identification capability.

The warehouse scene spans 72×75 square meters and consists of a large storage area with densely arranged shelves and goods, along with a smaller adjacent factory building. The repetitive structures and heavy occlusions make it challenging for robust perception and target localization.

Collectively, these environmental variations enable a comprehensive and highly rigorous evaluation of the agent's performance across diverse public real-world scenarios.

B.2 More Details of Outdoor Maps Construction in Carla

We use CARLA [12] to generate training data and evaluate our method across six maps. These maps include various layouts and paths to simulate the complexity of real-world urban scenes.

Town01 spans 66×361 square meters with a river and several bridges, challenging the robustness across varying features and colors.

Town02 spans 334×225 square meters with dense residential and commercial buildings. Its narrow sidewalks cause frequent occlusions, challenging obstacle avoidance and trajectory planning.

Town03 spans 585×547 square meters with large junctions and varied elevations, which cause significant perspective changes for tracking.

Town04 spans 716×554 square meters in a mountainous landscape with irregular paths. These terrain elevations cause frequent perspective changes during tracking.

Town05 spans 358×340 square meters, featuring a squared-grid layout with wide roads. Its open environment and straight paths provide the robot with a relatively clear line of sight.

Town10 spans 370×500 square meters, presenting a dense downtown environment with skyscrapers. Its severe background clutter and frequent pedestrian occlusions challenge tracking performance in highly dynamic environments.

B.3 More Details of Instruction Templates

We use a Vision-Language Model (VLM) to generate diverse tracking instructions from simulation images, avoiding repetitive command formats. For the appearance-based STT and DT tasks, the instruction generation prompt is defined as follows:

Instruction Generation Prompt

Describe the person directly in front of the camera in this simulation image. Identify their key visual characteristics (e.g., clothing color, type, accessories, patterns). Output exactly ONE concise sentence using one of these formats (choose randomly):

- 'Please follow the [features] [man/woman/person] ahead.'
- 'Follow the person [description] ahead.'
- 'Track the [man/woman] wearing [clothing details].'
- 'Stay behind the person in [outfit combination].'
- 'Go after the [person] with [distinctive features].'

In addition to VLM generation, we use a structured template strategy coupled with a synonym pool (e.g., `[ACTION]` $\in$ {"Follow", "Track", "Pursue", "Chase"}, `[ENTITY]` $\in$ {"person", "man", "woman"}) to systematically construct the three task levels:

STT & DT Task: Evaluates fundamental appearance-based visual grounding using macro-level attributes (e.g., gender, outfit style, dominant colors).

STT & DT Prompt

- *Format:* `[ACTION] + [ENTITY] + [MACRO_FEATURES]`
- *Examples:* "Track the woman wearing a dark gray armored outfit.", "Pursue the person in a dark top with a floral pattern."

AT Task: Focuses on pure spatial or temporal observation sequences without appearance descriptions. The model grounds the target based on egocentric cues or detection order.

AT Prompt

- *Format:* `[ACTION] + [ORDER] + [ENTITY] + [SPATIO-TEMPORAL_CUE]`
- *Examples:* "Stay behind the first person you detect.", "Follow the person directly in front of you."

Table 5: Comprehensive ablation studies on key hyperparameters. We analyze the box confidence threshold c_0, similarity threshold s_0, and the historical window size K on the challenging EVT-Bench Distracted Tracking (DT) task. **Bold** indicates the best performance, and default settings are marked with a light blue background.

Parameter	Value	SR ↑	TR ↑	CR ↓
Box Confidence Threshold (c_0)	0.85	51.2	65.5	5.3
	0.9	**51.5**	**65.8**	**5.1**
	0.95	51.4	65.7	5.2
Similarity Threshold (s_0)	0.88	50.7	65.0	5.7
	0.9	**51.5**	**65.8**	**5.1**
	0.92	51.0	65.2	5.5
Historical Window Size (K)	5	50.8	65.1	5.6
	6	**51.5**	**65.8**	**5.1**
	7	51.0	65.3	5.5

C Implementation Details and Extended Analysis

In this section, we provide the specific configurations of LT-VLA and present additional experimental results to further validate its robustness. Appendix C.1 details the hyperparameter settings and their sensitivity analysis, while Appendix C.2 provides a statistical evaluation of the model's stability across multiple runs.

C.1 Hyperparameter Details and Sensitivity Analysis

We evaluate the sensitivity of three key hyperparameters on the EVT-Bench DT task: box confidence threshold (c_0), similarity threshold (s_0), and historical window size (K). As shown in Tab. 5, the default settings are set to $c_0 = 0.9$, $s_0 = 0.9$, and $K = 6$.

Box Confidence Threshold (c_0). We test $c_0 \in \{0.85, 0.9, 0.95\}$. The performance remains highly stable across these values, indicating that our target localization is robust to the threshold.

Similarity Threshold (s_0). We evaluate $s_0 \in \{0.88, 0.9, 0.92\}$. The performance exhibits only marginal fluctuations. A lower (0.88) or higher (0.92) threshold causes minor drops of less than 1% in SR, indicating that the system is highly stable to similarity constraints.

Historical Window Size (K). We vary the window size $K \in \{5, 6, 7\}$. Similar to other parameters, the metrics remain highly consistent across different window sizes. The marginal performance differences confirm that our method is robust to the length of the historical context.

Overall, the results demonstrate that while LT-VLA exhibits strong robustness to parameter variations, the default configurations yield the highest tracking stability and safety.

C.2 Statistical Variance and Stability Analysis

To further validate the stability of LT-VLA, we conduct additional experiments on EVT-Bench across multiple runs with different random seeds to analyze the statistical variance. For the DT task, LT-VLA achieves an average Success Rate of $51.5 \pm 0.28\%$, a Tracking Rate of $65.8 \pm 0.42\%$, and a Collision Rate of $5.1 \pm 0.14\%$. The extremely low standard deviation (all $< 0.5\%$) across different random seeds confirms the reliability of our benchmark results.

D Limitations

Currently, the environments in LTEVT are constructed manually, which is resource-intensive and lacks direct real-world correspondence. In the future, we will explore the use of real-to-sim technologies for EVT data construction.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. **Claims**

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

   Answer: [Yes]

   Justification: See Section 1 for details.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 D for limitations.

Guidelines:

- The answer [N/A] 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: [N/A]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer [N/A] 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.1 for all information needed.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: The code and models are currently undergoing internal institutional review for intellectual property protection. We aim to release them as soon as the clearance is granted.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5.1 and Appendix C.1 for all the experimental details.

   Guidelines:

   - The answer [N/A] 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 Appendix C.2 for details.

   Guidelines:

   - The answer [N/A] 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 Section 5.1 for details.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] 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: We have strictly adhered to their respective terms of use.

    Guidelines:

    - The answer [N/A] 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: [No]

    Justification: The documentation is currently undergoing internal institutional review for intellectual property protection. We aim to release it as soon as the clearance is granted.

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

Guidelines:

- The answer [N/A] 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: [N/A]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: See Section 4.1 and Appendix B.3 for details.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.