TrackVLA: Embodied Visual Tracking in the Wild

Shaoan Wang[*,1,2] Jiazhao Zhang[*,1,2] Minghan Li[2] Jiahang Liu[2] Anqi Li[1,2]
Kui Wu[3] Fangwei Zhong[4] Junzhi Yu[†,1] Zhizheng Zhang[†,2,5] He Wang[†,1,2,5]

***Abstract*— Embodied visual tracking is a fundamental skill in Embodied AI, enabling an agent to follow a specific target in dynamic environments using only egocentric vision. This task is inherently challenging as it requires both accurate target recognition and effective trajectory planning under conditions of severe occlusion and high scene dynamics. Existing approaches typically address this challenge through a modular separation of recognition and planning. In this work, we propose TrackVLA, a Vision-Language-Action (VLA) model that learns the synergy between object recognition and trajectory planning. Leveraging a shared LLM backbone, we employ a language modeling head for recognition and an anchor-based diffusion model for trajectory planning. To train TrackVLA, we construct an Embodied Visual Tracking Benchmark (EVT-Bench) and collect diverse difficulty levels of recognition samples, resulting in a dataset of 1.7 million samples. Through extensive experiments in both synthetic and real-world environments, TrackVLA demonstrates SOTA performance and strong generalizability. It significantly outperforms existing methods on public benchmarks in a zero-shot manner while remaining robust to high dynamics and occlusion in real-world scenarios at 10 FPS inference speed. Our project page is: https://pku-epic.github.io/TrackVLA-web.**

I. INTRODUCTION

Embodied visual tracking (EVT) [1]–[6] requires the agent to persistently track a given target, which is a fundamental capability of embodied AI [7] and widely demanded in robotics [8], [9]. This task is particularly challenging due to its reliance on two tightly coupled skills: (1) Target recognition, the ability to accurately identify and distinguish the target, and (2) Trajectory planning, the capacity to determine optimal actions for effective tracking. The interplay between recognition and planning becomes especially demanding under challenging conditions, such as the presence of severe occlusion and highly dynamic scenes.

Toward achieving robust embodied visual tracking, existing methods [1]–[5], [10] typically address this challenge by decoupling recognition and trajectory planning into a detection model and a planning model, respectively. These approaches benefit from rapid advancements in visual foundation models [11]–[13] and policy learning techniques (*e.g.*, imitation learning [14] and reinforcement learning [3], [15]). Despite demonstrating early progress, these methods are limited to category-level tracking in relatively open areas. This is because their loosely coupled design causes error accumulation between the recognition model and the planning model—*e.g.*, an incorrect recognition may result in faulty planning, and vice versa.

[1]Peking University, [2]Galbot, [3]Beihang University, [4]Beijing Normal University, [5]BAAI, first author wangshaoan@stu.pku.edu.cn

Fig. 1: **TrackVLA** is a vision-language-action model capable of simultaneous object recognition and visual tracking, trained on a dataset of 1.7 million samples. It demonstrates robust tracking, long-horizon tracking, and cross-domain generalization across diverse challenging environments.

To achieve synergy between target recognition and trajectory planning, a versatile model must master both recognition and tracking capabilities. In this work, we propose TrackVLA, a vision-language-action model featuring a unified framework that integrates target recognition and trajectory planning. Specifically, both tasks utilize the same token encoding and LLM forwarding mechanism to predict the next token, while decoding is task-dependent. For the recognition task, TrackVLA employs a language modeling head to decode textual responses. For the planning task, TrackVLA leverages an anchor-based diffusion head to generate waypoint trajectories. Both tasks are trained jointly, optimizing TrackVLA to achieve tight coupling between recognition and planning.

To enable TrackVLA's acquisition of both recognition and planning capabilities, we collect 855K video recognition samples and 855K robot tracking samples. For recognition, we construct a human recognition dataset based on a public ReID dataset [16] and leverage open-world VQA datasets [17]–[19]. For embodied visual tracking data, we gather samples from a self-developed embodied visual tracking benchmark (EVT-Bench), which includes over 100 high-fidelity humanoid avatars moving randomly in simulated scenes. Both recognition and tracking samples were collected at varying difficulty levels to enable comprehensive training of TrackVLA.

We conduct extensive experiments on both synthetic and real-world environments, and we find that TrackVLA demonstrates superior performance with strong generalizability. TrackVLA archives SOTA performance in public benchmark Gym-UnrealCV [20] in a zero-shot manner, and significantly

outperforms baselines in self-built benchmark EVT-Bench that involves detailed language input and crowded environments. Furthermore, TrackVLA exhibits exceptional sim-to-real generalization capability, enabling robust tracking of previously unseen objects in novel environments at 10 FPS inference speed.

II. RELATED WORKS

The task requires agents to continuously pursue dynamic targets based on visual observations, relying on accurate target recognition and optimal trajectory planning. In real-world applications, human following [21]–[23] represents the most extensively studied scenario within this domain. While many recent works [9], [15], [24]–[29] decouple perception and planning into two separate modules—often incorporating visual foundation models [11] to enhance perception and employing reinforcement learning for planning—they frequently suffer from error accumulation due to the separation of detection and planning, as well as low training efficiency. To address this, some approaches leverage offline RL [6], [30] to boost training efficiency. However, the aforementioned approaches lack support for natural language inputs, which significantly limits their applicability in real-world human-robot interaction scenarios. To address this limitation, Uni-NaVid [14] introduced a vision-language-action (VLA) model that enables human following via large-scale imitation learning in simulation. Nonetheless, its reliance on a discrete action space hinders adaptability in complex, real-world environments. In contrast, TrackVLA integrates target recognition and trajectory planning into a unified training framework, achieving synergy between robust perception and flexible motion control, and demonstrating superior embodied visual tracking performance in real-world deployments.

III. METHOD

A. Embodied Visual Tracking Formulation

We formulate embodied visual tracking task as: At each timestamp T, given a natural language instruction $\mathscr{I}$, which describes the appearance of a specific target, and an egocentric RGB observation consisting of a sequence of frames $\mathscr{O}_T = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, the agent is required to output the next action $a_T \in \mathbb{A} = \{v, \omega\}$ to continuously follow the described target in unseen environments, where $\mathbb{A}$ is the action space including linear velocity v and angular velocity ω of the agent. The task is considered successful if the agent is able to consistently maintain an appropriate following distance (1–3 m) from the target while facing toward it.

B. TrackVLA Overview

As shown in Fig. 2, TrackVLA extends video-based VLM/VLA approaches [14], [31], [32] by introducing a parallel prediction branch for both trajectory planning and target recognition. For trajectory planning, TrackVLA organizes online-captured video data, combining historical and current observations, and concatenates them with tracking instructions and a special tracking token. A diffusion transformer then decodes the output tokens from a large language model

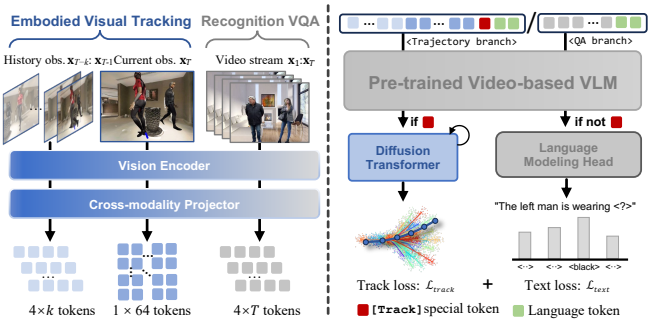


Fig. 2: **TrackVLA** is a vision-language-action model capable of simultaneous object recognition and visual tracking, trained on a dataset of 1.7 million samples. It demonstrates robust tracking, long-horizon tracking, and cross-domain generalization across diverse challenging environments.

(implemented with Vicuna-7B [33]) into waypoints. For recognition tasks, all video frames are encoded identically and processed in a conventional autoregressive manner.

C. TrackVLA Architecture

Observation Encoding. Given the egocentric RGB sequence $\mathscr{O}_T = \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, we employ a pre-trained vision encoder (EVA-CLIP [34]) to extract visual features $\mathbf{V}_{1:T} \in \mathbb{R}^{N \times C}$, where N is the number of patch (set to 256) and C represents the embedding dimension. Here, we apply a grid pooling strategy [14], [32] (Fig. 2, left) to the visual features, generating more compact representations. Specifically, we use two resolution scales:

$$\mathbf{V}^{\text{fine/coarse}} = GridPool(\mathbf{V}, \frac{64}{N} or \frac{4}{N}) \tag{1}$$

where $V_{\text{fine}} \in \mathbb{R}^{64 \times C}$ provides fine-grained observations, while $V_{\text{coarse}} \in \mathbb{R}^{4 \times C}$ offers coarse-grained observations. To optimally balance token length and performance, we empirically use fine-grained features $V_{\text{fine}} \in \mathbb{R}^{64 \times C}$ for the latest tracking observation to enhance target identification, while coarse-grained tokens are used for historical tracking and VQA-based recognition.

To ensure consistent inference speed during tracking, we employ a sliding window mechanism to retain only the latest k frames (set to 32 in our implementation). For embodied visual tracking, we structure the visual token sequence as: $\mathscr{V}_T^{\text{track}} = \{\mathbf{V}_{T-k}^{\text{coarse}}, ..., \mathbf{V}_{T-1}^{\text{coarse}}, \mathbf{V}_T^{\text{fine}}\}$, while for the video question answering (VQA) recognition task, we construct the sequence as: $\mathscr{V}_T^{\text{VQA}} = \{\mathbf{V}_1^{\text{coarse}}, ..., \mathbf{V}_T^{\text{coarse}}\}$. Following established Vision-Language Models (VLMs) [31], [35], we use a cross-modality projector $\mathscr{P}(\cdot)$ (a 2-layer MLP) to project visual features into the latent space of the Large Language Model: $\mathbf{E}_T^V = \mathscr{P}(\mathscr{V}_T)$.

Large Language Model Forwarding. We concatenate the visual tokens $\mathbf{E}_T^V$ with the language tokens $\mathbf{E}^I$ (adding a special `[Track]` token for the tracking task) and feed them into the LLM (Fig. 2 Right) to obtain the predicted token $\mathbf{E}_T^{\text{pred}}$. The predicted token is then processed differently depending on the task (determined by the presence of the `[Track]` token). For recognition tasks, we use the standard language

modeling head to decode the token auto-regressively into a vocabulary word [36]. For tracking tasks, $\mathbf{E}_T^{\text{pred}}$ serves as conditional input to our action head model, which generates waypoint trajectories for navigation.

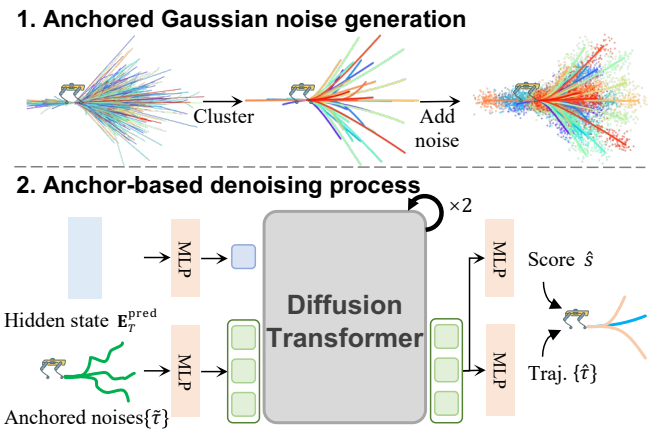


Fig. 3: Anchor-based Diffusion Action Model.

Anchor-based Diffusion Action Model. We employ an anchor-based diffusion model [37] that performs denoising from predefined anchors to generate waypoints. These predefined anchors provide initial coarse trajectories that significantly reduce the required denoising iterations, yielding a 5× speedup compared to vanilla diffusion policies [38], [39]. As shown in Fig. 3, we first collect all trajectories from the training data and apply K-means clustering [40] to obtain a set of trajectory anchors $\{\tau_i\}_{i=1}^{M}$, where M denotes the number of anchors. Each anchor $\tau_i = (x_i, y_i, \theta_i)_{i=1}^{N_w}$ represents a robot trajectory pattern, where N_w is the number of waypoints in each trajectory. We then perturb each anchor with Gaussian noise to create noised anchors $\{\tilde{\tau}_i\}_{i=1}^{M}$. Our action model $\mathscr{A}_\theta(\cdot)$ takes the set of noised anchors $\{\tilde{\tau}_i\}_{i=1}^{M}$ and the condition $\mathbf{E}_T^{\text{pred}}$ as input, and outputs: the denoised trajectories $\{\hat{\tau}_i\}_{i=1}^{M}$ and the corresponding trajectory classification scores $\{\hat{s}_i\}_{i=1}^{M}$:

$$\{\hat{s}_i, \hat{\tau}_i\}_{i=1}^{M} = \mathscr{A}_\theta\left(\{\tilde{\tau}_i\}_{i=1}^{M}, \mathbf{E}_T^{\text{pred}}\right) \tag{2}$$

For each sample, we label the anchor trajectory closest to the ground truth τ_{gt} as positive ($s_{\text{nearest}} = 1$), and all others as negative ($s_{\text{else}} = 0$). We then jointly optimize the trajectory regression loss and the score prediction loss. The tracking loss $\mathscr{L}_{track}$ is defined as:

$$\mathscr{L}_{\text{track}} = \sum_{i=1}^{M} [s_i MSE(\hat{\tau}_i, \tau_{gt}) + \lambda BCE(\hat{s}_i, s_i)] \tag{3}$$

where λ is a balancing parameter. Here, we adopt the Diffusion Transformer (DiT) [41] for denoising and the anchor-based diffusion policy only needs two denoising steps. Given a batch of input sequences, the overall training loss $\mathscr{L}$ is defined as a weighted combination of the tracking loss $\mathscr{L}_{track}$ and the text prediction loss $\mathscr{L}_{text}$, formulated as $\mathscr{L} = \mathscr{L}_{track} + \alpha \mathscr{L}_{text}$, where α is also a balancing parameter. More details can be found in the Appendix.

D. Implementation Details

Training Details. During training, we follow the standard practice in vision-language modeling (VLM) [35] by training for only one epoch. Additionally, we freeze the parameters of the vision encoder throughout training. **Inference Details.** During inference, we use a special token `[Track]` to indicate the current task. When the `[Track]` token is present, LLM performs only a single-step autoregression and passes the output hidden state to the action model to predict trajectories. We apply the DDIM [42] update rule for denoising with only two steps, and select the trajectory $\hat{\tau}_k$ corresponding to the top-1 score $\hat{s}_k$ as the final output. Otherwise, LLM conducts full autoregressive decoding to answer the given question based on visual observations. More details are provided in the Appendix.

IV. Data Collection

To train our parallel branch TrackVLA, we collect both embodied visual tracking data (855K samples) and video-based question-answering data (855K samples). For tracking samples (Sec. IV-A), we develop a custom avatar-following simulator and collect a diverse dataset spanning challenging scenarios. For recognition samples (Sec. IV-B), we construct a video question-answering dataset that requires the agent to describe or distinguish target objects amidst complex backgrounds and distractors.

A. Embodied Visual Tracking Data

Embodied Visual Tracking Simulator. We build our embodied visual tracking simulator based on Habitat 3.0 [9], which provides an off-the-shelf simulation engine for collision detection and rendering. Our main enhancements include two aspects: (1) ***Humanoid Avatar Generation.*** We implement a fully automated pipeline for generating and annotating diverse humanoid avatars (Fig. 4 (A)). Specifically, we adopt the SMPL-X human model and initialize the avatars with random shapes and randomly sampled UV texture maps (ATLAS dataset [43]). We then use a vision-language model (Qwen-VL2.5 [44]) to obtain corresponding textual descriptions of the avatars. (2) ***Natural Human Behaviors.*** We assign each avatar a series of targets that it must reach in order, with on-and-off walking states. The walking speed is randomly sampled from a natural human walking speed range of [1.0 m/s - 1.5 m/s] [45]. Furthermore, we employ the ORCA algorithm [46] to enable dynamic collision avoidance and responsive interactions, resulting in more natural behavior. For more details, please refer to the Appendix.

Embodied Visual Tracking Benchmark. Based on our simulator, we construct the Embodied Visual Tracking Benchmark (EVT-Bench) to comprehensively evaluate embodied visual tracking capabilities. We generate 100 diverse humanoid avatars and corresponding descriptions and utilize 804 scene environments from HM3D [47] and MP3D [48]. A total of 25,986 episodes are generated and subsequently divided into training and testing sets, ensuring no overlap of avatars or scenes between the two splits. The training

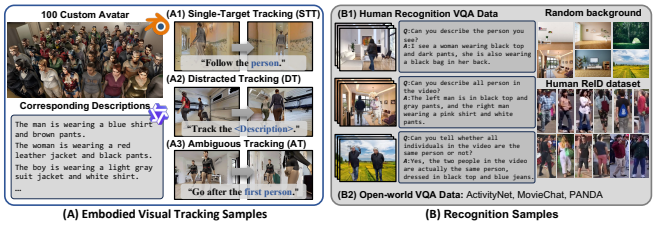


Fig. 4: **Overview of the training datasets used in TrackVLA.** We collect 855K embodied visual tracking samples and 855K open-world recognition samples to jointly enhance the robust recognition and tracking capabilities of TrackVLA.

set consists of 21,771 episodes across 703 scenes, while the testing set includes 4,215 episodes across 101 unseen scenes. To comprehensively evaluate algorithm performance across different scenarios, EVT-Bench is divided into three sub-task categories of increasing difficulty. Each sub-task contains 7,257 episodes for training and 1,405 episodes for testing.

Tracking Data Collection. We collect 885K embodied visual tracking samples in the EVT-Bench training split, covering three sub-tasks of varying difficulty. Each sample includes a navigation history (RGB sequence), a target description, and the corresponding expert trajectory τ_{gt}. Additional details regarding the benchmark and data collection are provided in the Appendix. The EVT-Bench will be made publicly available to benefit the research community.

B. Video Question Answering Dataset

Despite our considerable efforts to incorporate diverse avatars and indoor scenes, the tracking samples remain limited to synthetic environments. To equip TrackVLA with open-world recognition capabilities (beyond tracking samples), we further collect a total of 855K recognition samples and jointly train them with the tracking samples. Specifically, the recognition video question-answering (VQA) samples consist of 362K human recognition samples and 493K open-world VQA samples.

For the human recognition VQA data, we leverage SYNTH-PEDES [16], a large-scale person-text dataset, to construct VQA samples that require TrackVLA to identify or describe individuals in videos featuring randomly composed human subjects and background scenes. Each sample is created by placing 1–3 randomly selected human images onto diverse backgrounds, with accompanying textual descriptions detailing each individual's attributes, their relative spatial positions, and whether they represent the same identity. In addition to human recognition samples, we also incorporate publicly available VQA samples [17]–[19] that provide open-world captions. These samples enhance TrackVLA's ability to recognize open-world targets.

V. Experiments

Performance on EVT-Bench. We further evaluate our method on the proposed benchmark, EVT-Bench, as shown in Table I. TrackVLA significantly outperforms existing approaches across all three tasks (`STT`, `DT`, and `AT`), demonstrating its robust and comprehensive tracking capabilities, particularly in comparison to the VLA method

Methods	*STT* SR↑/ TR↑ / CR↓	*DT* SR↑/ TR↑ / CR↓	*AT* SR↑/ TR↑ / CR↓
IBVS† [49]	42.9/56.2/3.75	10.6/28.4/6.14	15.2/39.5/**4.90**
PoliFormer† [26]	4.67/15.5/40.1	2.62/13.2/44.5	3.04/15.4/41.5
EVT [6]	24.4/39.1/42.5	3.23/11.2/47.9	17.4/21.1/45.6
EVT‡ [6]	32.5/49.9/40.5	15.7/35.7/53.3	18.3/21.0/44.9
Uni-NaVid [14]	25.7/39.5/41.9	11.3/27.4/43.5	8.26/28.6/43.7
Ours	**85.1/78.6/1.65**	**57.6/63.2/5.80**	**50.2/63.7**/17.1

TABLE I: Performance on EVT-Bench. †: Use GroundingDINO [13] as the open-vocabulary detector. ‡: Use SoM [50]+GPT-4o [51] as the visual foundation model.

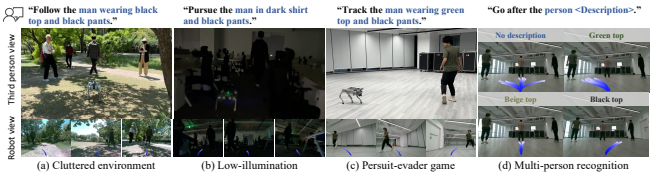


Fig. 5: **Real-world qualitative results of TrackVLA.** TrackVLA is deployed in a zero-shot manner across diverse environments, executing diverse tracking instructions in challenging scenarios.

Uni-NaVid [14]. However, despite these improvements, we observe a noticeable performance drop when transitioning from single-target tracking (STT) to distracted tracking (DT) and ambiguity tracking (AT), highlighting the challenges of accurately recognizing and following a specified target in complex environments with distractors. We believe our benchmark can benefit the research community by providing a well-defined target for future studies.

A. Qualitative Results in Real-World

We provide qualitative real-world results in Fig. 5, where we evaluate our method in challenging scenarios, including: (A) cluttered environments, (B) low-lighting conditions, (C) pursuit-evasion tasks, and (D) multi-person recognition. The experimental results demonstrate that TrackVLA exhibits strong sim-to-real transfer capabilities in both recognition and tracking, while maintaining high-frequency inference in real-world scenarios, thereby enabling zero-shot deployment in highly dynamic environments. For additional real-world performance demonstrations, we refer the audience to our supplementary video.

VI. Conclusions

In this work, we propose TrackVLA, a Vision-Language-Action (VLA) model designed for the embodied visual tracking task. TrackVLA supports the output of both tracking trajectories and text-based responses. It is jointly trained on both embodied visual tracking data and open-world recognition data, enabling it to learn the synergy between these two modalities. To support this, we collect a large-scale dataset consisting of 855K embodied visual tracking samples and 855K open-world recognition samples. Extensive experiments demonstrate its state-of-the-art performance in simulation and strong generalization, enabling zero-shot deployment in real-world scenarios.

REFERENCES

[1] A. Maalouf, N. Jadhav, K. M. Jatavallabhula, M. Chahine, D. M. Vogt, R. J. Wood, A. Torralba, and D. Rus, "Follow anything: Open-set detection, tracking, and following in real-time," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3283–3290, 2024. 1

[2] W. Zhang, K. Song, X. Rong, and Y. Li, "Coarse-to-fine uav target tracking with deep reinforcement learning," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1522–1530, 2018. 1

[3] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang, "Ad-vat: An asymmetric dueling mechanism for learning visual active tracking," in *International Conference on Learning Representations*, 2019. 1, 7

[4] ——, "Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1467–1482, 2019. 1, 8

[5] F. Zhong, X. Bi, Y. Zhang, W. Zhang, and Y. Wang, "Rspt: reconstruct surroundings and predict trajectory for generalizable active object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3705–3714. 1, 8

[6] F. Zhong, K. Wu, H. Ci, C. Wang, and H. Chen, "Empowering embodied visual tracking with visual foundation models and offline rl," in *European Conference on Computer Vision*. Springer, 2024, pp. 139–155. 1, 2, 4, 7, 8

[7] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 230–244, 2022. 1

[8] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023. 1

[9] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," *arXiv preprint arXiv:2310.13724*, 2023. 1, 2, 3

[10] J. Li, J. Xu, F. Zhong, X. Kong, Y. Qiao, and Y. Wang, "Pose-assisted multi-camera collaboration for active object tracking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 759–766. 1

[11] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026. 1, 2

[12] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. 1

[13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023. 1, 4

[14] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, "Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks," *arXiv preprint arXiv:2412.06224*, 2024. 1, 2, 4, 8

[15] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking via reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 3286–3295. 1, 2

[16] J. Zuo, J. Hong, F. Zhang, C. Yu, H. Zhou, C. Gao, N. Sang, and J. Wang, "Plip: Language-image pre-training for person representation learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 45 666–45 702, 2024. 1, 4

[17] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang *et al.*, "Moviechat: From dense token to sparse memory for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 221–18 232. 1, 4

[18] T.-S. Chen, A. Siarohin, W. Menapace, E. Deyneka, H.-w. Chao, B. E. Jeon, Y. Fang, H.-Y. Lee, J. Ren, M.-H. Yang *et al.*, "Panda-70m: Captioning 70m videos with multiple cross-modality teachers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 320–13 331. 1, 4

[19] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "Activitynet-qa: A dataset for understanding complex web videos via question answering," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9127–9134. 1, 4

[20] W. Qiu, F. Zhong, Y. Zhang, S. Qiao, Z. Xiao, T. S. Kim, and Y. Wang, "Unrealcv: Virtual worlds for computer vision," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1221–1224. 1

[21] H. Ye, J. Zhao, Y. Zhan, W. Chen, L. He, and H. Zhang, "Person re-identification for robot person following with online continual learning," *IEEE Robotics and Automation Letters*, 2024. 2

[22] H. Ye, K. Cai, Y. Zhan, B. Xia, A. Ajoudani, and H. Zhang, "Rpf-search: Field-based search for robot person following in unknown dynamic environments," *arXiv preprint arXiv:2503.02188*, 2025. 2

[23] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *ACM Transactions on Human-Robot Interaction*, vol. 14, no. 2, pp. 1–65, 2025. 2

[24] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 6, pp. 1317–1332, 2019. 2, 7

[25] A. Devo, A. Dionigi, and G. Costante, "Enhancing continuous control of mobile robots for end-to-end visual active tracking," *Robotics and Autonomous Systems*, vol. 142, p. 103799, 2021. 2

[26] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, "Poliformer: Scaling on-policy rl with transformers results in masterful navigators," *arXiv preprint arXiv:2406.20083*, 2024. 2, 4, 8

[27] F. Zhong, P. Sun, W. Luo, T. Yan, and Y. Wang, "Towards distraction-robust active visual tracking," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 782–12 792. 2

[28] A. Bajcsy, A. Loquercio, A. Kumar, and J. Malik, "Learning vision-based pursuit-evasion robot policies," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 9197–9204. 2

[29] L. Scofano, A. Sampieri, T. Campari, V. Sacco, I. Spinelli, L. Ballan, and F. Galasso, "Following the human thread in social navigation," *arXiv preprint arXiv:2404.11327*, 2024. 2

[30] D. Shah, A. Bhorkar, H. Leen, I. Kostrikov, N. Rhinehart, and S. Levine, "Offline reinforcement learning for visual navigation," *arXiv preprint arXiv:2212.08244*, 2022. 2

[31] Y. Li, C. Wang, and J. Jia, "Llama-vid: An image is worth 2 tokens in large language models," *arXiv preprint arXiv:2311.17043*, 2023. 2

[32] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "Navid: Video-based vlm plans the next step for vision-and-language navigation," *Robotics: Science and Systems*, 2024. 2

[33] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 2, 6

[34] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv preprint arXiv:2303.15389*, 2023. 2, 6

[35] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023. 2, 3, 6

[36] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023. 3

[37] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang *et al.*, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," *arXiv preprint arXiv:2411.15139*, 2024. 3

[38] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70. 3

[39] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023. 3

[40] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006. 3

[41] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205. 3
[42] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020. 3
[43] Y. Liu, J. Zhu, J. Tang, S. Zhang, J. Zhang, W. Cao, C. Wang, Y. Wu, and D. Huang, "Texdreamer: Towards zero-shot high-fidelity 3d human texture generation," in *European Conference on Computer Vision*. Springer, 2024, pp. 184–202. 3
[44] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025. 3
[45] R. L. Knoblauch, M. T. Pietrucha, and M. Nitzburg, "Field studies of pedestrian walking speed and start-up time," *Transportation research record*, vol. 1538, no. 1, pp. 27–38, 1996. 3
[46] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19. 3
[47] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 3
[48] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niebner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 667–676. 3
[49] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1415–1427, 2016. 4, 8
[50] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao, "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v," *arXiv preprint arXiv:2310.11441*, 2023. 4, 9
[51] OpenAI, "Introducing 4o image generation," https://openai.com/index/introducing-4o-image, 2024, accessed: 2025-04-29. 4, 9
[52] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6182–6191. 7
[53] Q. Jiang, L. Wu, Z. Zeng, T. Ren, Y. Xiong, Y. Chen, Q. Liu, and L. Zhang, "Referring to any person," *arXiv preprint arXiv:2503.08507*, 2025. 8
[54] S. Yang, T. Qu, X. Lai, Z. Tian, B. Peng, S. Liu, and J. Jia, "Lisa++: An improved baseline for reasoning segmentation with large language model," *arXiv preprint arXiv:2312.17240*, 2023. 9

APPENDIX

VII. Training Details

Similar to conventional vision-language models (VLMs), TrackVLA follows a two-stage training pipeline. In the first stage, we train the projector of the visual encoder using a large amount of image-caption data [35] to align the visual embedding space with the LLM's latent space. In the second stage, we jointly train the visual projector, the large language model, and the action model using a mixture of the training data. During training, we truncate the diffusion schedule of the action model to at most 50 out of a total of 1000 steps to diffuse the trajectory anchors, which introduces only a small amount of noise.

TrackVLA is trained on a cluster server equipped with 24 NVIDIA H100 GPUs for approximately 15 hours, totaling 360 GPU hours. The vision encoder (EVA-CLIP [34]) and the large language model (Vicuna-7B [33]) are initialized with their respective pretrained weights, and the vision encoder remains frozen throughout the entire training process. Following standard VLM practices, we train the model for only one epoch. The training is conducted with a learning rate of 2e-5, a total batch size of 196, and a cosine learning rate schedule with linear warm-up. We use the AdamW optimizer for optimization. See Table II for detailed parameter settings.

VIII. Inference Details

During inference, each input frame is resized to 224×224 and fed into the vision encoder. After obtaining visual tokens, we organize the tokens according to the task type. For the embodied visual tracking task, we prepend a special `[Track]` token before the instruction tokens and perform only a single-step autoregression with the LLM. The final-layer hidden state output from the LLM is then passed to the action model. We apply 10 out of 1000 diffusion steps to the trajectory anchors and use DDIM to perform 2 denoising steps, resulting in a set of predicted trajectories and corresponding score vectors. The trajectory corresponding to the anchor with the top 1 score is selected as the final output. For the VQA task, we follow the standard autoregressive decoding process of the LLM, and the language modeling head detokenizes the predicted tokens into textual answers. See Table II for detailed parameter settings.

IX. EVT-Bench

A. Episode Generation

For each episode, we first sample a motion trajectory for the target humanoid avatar within the navigable area. Each trajectory consists of a start point, a random number of intermediate waypoints (0–2), and an end point. The distance between any two consecutive waypoints must exceed a predefined minimum threshold $d_{min} = 3$ m. After generating the trajectory for the target avatar, the agent is placed near the target's starting point, with its initial orientation roughly facing the target but perturbed by a random offset within 30°. For the *DT* and *AT* tasks, distractors are initialized near the target's trajectory, and their paths are designed to intersect with the target's trajectory as much as possible to enhance the level of distraction.

B. Evaluation

In each episode, the target humanoid and distractors move along their predefined trajectories. The evaluated algorithm receives the agent's observation at each time step and performs inference to generate the corresponding control command, consisting of linear and angular velocities of the agent. The agent then moves according to the speed command. The episode terminates when the target humanoid reaches its destination or when the agent collides with the humanoid.

C. Metric Definitions

- **Success Rate (SR)**: This metric evaluates the agent's tracking ability. An episode is considered successful if, by its end, the agent remains oriented toward the target and maintains a safe distance of 1–3 meters. The success

Notation	Shape & Params.	Description
lr	2e-5	learning rate
B	196	batch size
T	1000	total diffusion steps
T_{train}	50	number of noise addition steps during training
T_{infer}	10	number of noise addition steps during training
N_{step}	2	denoising steps during inference
$\mathbf{X}$	224×224	input observation size
N	256	the number of image patch
C	1408	embedding dimension of visual feature
α	1	balancing parameter 1
λ	100	balancing parameter 2
M	40	the number of trajectory anchors
N_w	10	the number of waypoints

TABLE II: Hyperparameters and notation used in our model.

rate is defined as the proportion of successful episodes over the total number of episodes.

- **Tracking Rate (TR)**: This metric evaluates the tracking quality of the aggent. It is defined as the proportion of steps S where the agent successfully tracks the target to the total number of steps L, i.e., $\text{TR} = S/L$.
- **Collision Rate (CR)**: This metric evaluates the safety of the agent. It is defined as the proportion of episodes that terminate due to a collision between the agent and the target humanoid avatar.

D. Humanoid Avatar Gallery

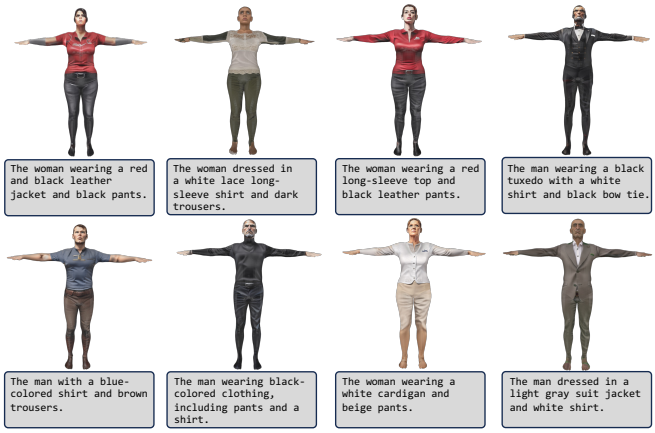


Fig. 6: Visualization of the custom humanoid avatars with captions.

We provide visualizations of several custom-designed humanoid avatars paired with their descriptive captions, as shown in Fig. 6

E. Visualization of Training Data

We provide visualizations of the training set of self-built EVT-Bench, as shown in Fig. 10

F. Qualitative Results on EVT-Bench

We provide visual results of TrackVLA on EVT-Bench, shown in Fig. 11.

X. Detailed Experiments of Gym-UnrealCV

A. Evaluation

The evaluation setting of Gym-UnrealCV follows [6], where each episode has a maximum length of 500 steps. The agent's tracking region is defined as a 90-degree fan-shaped sector with a radius of 750 cm. Success is achieved if the agent keeps the target within this region for the entire episode. A failure occurs if the target remains outside the region for more than 50 consecutive steps.

B. Metric Definitions

- **Episode Length (EL)**: The average number of steps per episode over 100 episodes, reflecting the model's long-term tracking capability under predefined termination conditions.
- **Success Rate (SR)**: The percentage of successful episodes out of the total 100 episodes, measuring the model's robustness in active visual tracking.

C. Testing Scenes

- **SimpleRoom**: A basic environment designed to verify the model's fundamental tracking capability.
- **Parking Lot**: An environment featuring occlusions and low-light conditions.
- **UrbanCity**: A typical urban street scene with reflective road surfaces.
- **UrbanRoad**: Similar to UrbanCity with fewer obstacles.
- **Snow Village**: A challenging terrain with uneven surfaces and complex backlighting.

D. Baselines

- **DiMP** [52]: Utilizes a pre-trained video tracker to generate target bounding boxes as scene representations and applies a PID controller for motion control.
- **SARL** [24]: An online reinforcement learning (RL) approach that encodes RGB observations into latent visual features and trains an end-to-end Conv-LSTM policy via RL.
- **AD-VAT** [3]: Introduces an asymmetric dueling mechanism and trains an RL-based tracker with a learnable adversarial target to improve robustness.

Methods	SimpleRoom	Parking Lot	UrbanCity	UrbanRoad	Snow Village	Mean
DiMP	**500/1.00**	327/0.48	401/0.66	308/0.33	301/0.43	367/0.58
SARL	**500/1.00**	301/0.22	471/0.86	378/0.48	318/0.31	394/0.57
AD-VAT	**500/1.00**	302/0.20	484/0.88	429/0.60	364/0.44	416/0.62
AD-VAT+	**500/1.00**	439/0.60	497/0.94	471/0.94	365/0.44	454/0.76
TS	**500/1.00**	472/0.89	496/0.94	480/0.84	424/0.63	474/0.86
RSPT	**500/1.00**	480/0.80	**500/1.00**	**500/1.00**	410/0.80	478/0.92
EVT	**500/1.00**	484/0.92	**500/1.00**	496/0.96	471/0.87	490/0.95
Ours	**500/1.00**	**500/1.00**	**500/1.00**	**500/1.00**	**500/1.00**	**500/1.00**

TABLE III: **Quantitative results compared with baselines in unseen environments.** The two metrics of each cell represent the Average Episode Length (EL) and Success Rate (SR).

- **AD-VAT+** [4]: An enhanced version of AD-VAT that incorporates a two-stage training scheme, aiming to improve performance in cluttered and obstacle-rich environments.
- **TS** [5]: A teacher-student framework that extends behavior cloning by leveraging a pose-based teacher to provide real-time supervision for the vision-based student policy during interaction.
- **EVT** [6]: An offline RL-based framework designed for dynamic target following, which integrates vision foundation models to enhance perception and robustness.
- **IBVS** [49]: A model-based method that takes the target bounding box as input and applies a Kalman filter-based visual servoing algorithm to follow the target.
- **PoliFormer** [26]: A reinforcement learning-based navigation framework that explicitly encodes target bounding boxes into the observation space to enhance tracking accuracy.
- **Uni-NaVid** [14]: A unified vision-language-action (VLA) model designed for general navigation tasks, including human following.

E. Experiment Results

Single Human Tracking Evaluation The single human tracking task spans five distinct environments mentioned above, covering a wide range of variations in lighting conditions, viewpoints, and scene layouts. As shown in Table III, TrackVLA achieves state-of-the-art performance across all five environments and successfully passes all test cases. Notably, TrackVLA is trained without any data from this simulator, highlighting its strong generalization under a **zero-shot** transfer setting.

Distraction Robustness Evaluation In this experiment, distractors with appearances identical to the target are introduced, requiring the agent to consistently track the first observed target. TrackVLA addresses this challenge using instructions such as "Follow the first person you see". Experimental results in Table IV show that TrackVLA achieves state-of-the-art performance across all scenarios, demonstrating its strong capability in understanding and reasoning about human motion.

Unseen Object Generalization Evaluation We further evaluate the object-level generalization ability of TrackVLA using the Gym-UnrealCV benchmark. Specifically, in the *SimpleRoom* environment, we test the model's tracking performance on four unseen animal categories: horse, dog, sheep, and pig. As shown in Table V, TrackVLA successfully tracks all four categories, consistent with its performance on the single-person tracking task. This demonstrates its strong generalization capability to novel object types.

Methods	Parking Lot (2D)	UrbanCity (4D)	ComplexRoom (4D)
DiMP	271/0.24	348/0.32	307/0.26
SARL	237/0.12	221/0.16	263/0.15
AD-VAT	232/0.13	204/0.06	223/0.16
AD-VAT+	166/0.08	245/0.11	262/0.18
TS	331/0.39	381/0.51	401/0.54
EVT	425/0.63	472/**0.92**	**479**/0.88
Ours	**467/0.90**	**476/0.92**	**479/0.91**

TABLE IV: **Evaluating the distraction robustness in the environment with distractors.** (4D) represents that there are 4 distractors in the environment.

Methods	Horse	Dog	Sheep	Pig
EVT	**500/1.00**	469/0.90	471/0.93	472/0.94
Ours	**500/1.00**	**500/1.00**	**500/1.00**	**500/1.00**

TABLE V: **Evaluating the generalization on the unseen category of the target in SimpleRoom.** We directly adopt the agent on the unseen animals: horse, dog, sheep, and pig.

F. Visualization of Humanoid Avatar in Gym-UnrealCV

We showcase several humanoid avatars used in Gym-UnrealCV in Fig. 7.

Fig. 7: Examples of humanoid avatars used in Gym-UnrealCV.

G. Qualitative Results on Gym-UnrealCV

We provide visual results of TrackVLA on Gym-UnrealCV, shown in Fig. 12.

XI. Visual Recognition Experiment

A. Baselines

- **RexSeek** [53]: A Multimodal Large Language Model (MLLM) designed to detect people or objects in images based on natural language descriptions.

- **LISA++** [54]: A Multimodal Large Language Model capable of both language understanding and mask generation.
- **SoM+GPT-4o** [50], [51]: A visual prompting method that guides large multimodal models like GPT-4o to perform visual grounding by overlaying segmented image regions with identifiable marks.

B. Evaluation

During testing, each test image contains two *unseen* persons positioned on the left and right sides, and two corresponding descriptions are provided for each person. Given the differing output formats of the evaluated methods, we define task-specific evaluation criteria. RexSeek is an object detection model that outputs bounding boxes; we evaluate its performance by checking whether the predicted box correctly selects the target person. LISA++ is an instance segmentation model that outputs a mask for the target; we assess whether the mask covers the correct individual. The SoM+GPT-4o pipeline first performs image segmentation, then uses SoM to overlay numerical marks on the original image at the location of each segmentation mask. The annotated image is then passed to GPT-4o, which selects the number that best matches the given description. For this method, we evaluate whether the mask corresponding to the selected mark covers the correct individual. As for TrackVLA, which outputs a future trajectory, we determine correctness by checking whether the trajectory direction aligns with the corresponding target person.

XII. More Ablation Study

A. Action Model Architecture

The architectures evaluated in this ablation study include Multi-Layer Perceptrons (MLPs) with 3 and 6 layers, respectively, as well as diffusion transformers of varying scales. The hidden state dimensions for the two MLPs are set to 1024 and 4096. The base diffusion transformer is configured with a depth of 12, hidden size of 768, and 12 attention heads, while the small diffusion transformer uses a depth of 6, hidden size of 384, and 4 attention heads.

Model	Params.	SR↑	TR↑	CR↓	time(ms) ↓
Autoregressive	131M	42.6	56.9	11.7	460
MLP (3-Layers)	7M	45.8	59.9	10.1	**0.5**
MLP (6-Layers)	89M	52.7	61.9	9.42	0.8
DP-Base	89M	17.9	33.8	27.7	65
Ours-Small	13M	49.8	60.2	6.67	8
Ours-Base	89M	**57.6**	**63.2**	**5.80**	13

TABLE VI: Comparison of different action models.

B. History Window Length

Incorporating historical observations helps the model better infer the target's motion pattern and relative position. Here, we investigate how varying the length of the history observation window L_{his} affects model performance. Table VII shows that removing history observations leads to a significant performance drop. We empirically select 32 as the optimal window length.

L_{his}	SR↑	TR↑	CR↓
0	29.9	49.6	6.94
32	**57.6**	63.2	**5.80**
64	56.5	**63.3**	6.49

TABLE VII: Comparison of different history window lengths.

C. Future Trajectory Horizon

TrackVLA predicts a future trajectory consisting of L_{traj} waypoints. In Table VIII, we investigate the impact of varying the number of predicted waypoints on overall performance. Experimental results show that using 10 waypoints yields the best performance.

L_{traj}	SR↑	TR↑	CR↓
1	44.3	60.6	14.4
10	**57.6**	**63.2**	**5.80**
20	51.3	60.2	7.54

TABLE VIII: Comparison of different predicted waypoint lengths.

D. Human Recognition Dataset

Furthermore, we investigate the impact of different types of human recognition data on the model's recognition capability. Specifically, we categorize the data into three types: Single Human, Multiple Human, and Same Human, corresponding to images containing one person, 2–3 different individuals, and two identical individuals, respectively. For each category, we construct dedicated human recognition datasets and evaluate the model's recognition performance under each data setting. Table IX presents the model's recognition performance under different types of human recognition data. The experimental results demonstrate that the inclusion of each type of human recognition data leads to improved model recognition performance.

In addition, we conduct another analysis to evaluate the impact of removing random backgrounds by replacing all the human recognition data with a plain white background. As presented in Table IX, removing the random background leads to a notable performance drop.

Single Human	Multiple Human	Same Human	Random Background	ACC↑	ACC Drop
✗	✗	✗	✗	62.0	22.9% ↓
✓	✗	✗	✓	72.3	10.4% ↓
✓	✓	✗	✓	76.7	4.60% ↓
✓	✓	✓	✗	67.4	16.2% ↓
✓	✓	✓	✓	**80.7**	-

TABLE IX: Comparison of different human recognition data.

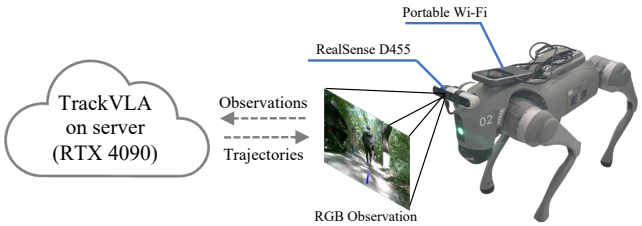


Fig. 8: **Real-world system architecture.** TrackVLA is deployed on a remote server, and the robot communicates with it via the Internet.

XIII. Real-world Deployment

A. Robot Platform

We provide a visualization of our robotic platform in Fig. 8. The platform is based on the Unitree GO2 quadruped robot, equipped with an Intel RealSense D455 camera. In our work, only RGB frames with a resolution of 640×480 are utilized, under a horizontal field of view (HFOV) of 90°. Additionally, a portable Wi-Fi is mounted on the back of the robot to enable communication with the remote server through the Internet.

B. Real-world System Architecture

TrackVLA is deployed on a remote server equipped with an NVIDIA RTX 4090 GPU. During tracking, the server receives the instructions and images captured by the Intel RealSense D455 camera via the Internet. To ensure efficient communication, the images are compressed before transmission. After processing the incoming images, the model performs inference and predicts the future trajectory, which is then transmitted to the quadruped robot for execution. Upon receiving the predicted trajectory, the robot employs a pure pursuit algorithm, combined with its pose information, to perform closed-loop control of its linear and angular velocities, enabling it to follow the trajectory accurately. Additionally, the robot leverages LiDAR point cloud data and implements an elastic band algorithm to achieve obstacle avoidance.

XIV. Real-world Experiments

To further evaluate the tracking capability of TrackVLA, we conducted extensive real-world experiments comparing a quadruped robot powered by TrackVLA with a leading commercial tracking drone (DJI Flip). We tested the following three levels of tracking scenarios with increasing difficulty, each repeated 10 times:

- *Easy*: tracking in open outdoor environments without obstacles;
- *Medium*: tracking in complex environments with occlusions such as walls;
- *Hard*: tracking a target moving at high speed.

The results are shown in Table X. Both TrackVLA and DJI Flip achieved a 100% success rate in the *Easy* setting. However, as task difficulty increased, the performance of DJI Flip dropped significantly, falling well below that of TrackVLA. Figure 9 further illustrates several representative cases where TrackVLA succeeded while DJI Flip failed. Additional details of the real-world experiments are provided in the supplementary video.

Method	Easy	Medium	Hard
DJI Flip	**100%**	70%	50%
TrackVLA	**100%**	**90%**	**70%**

TABLE X: **Real-world tracking experiments.** We compare TrackVLA with the commercial tracking drone.

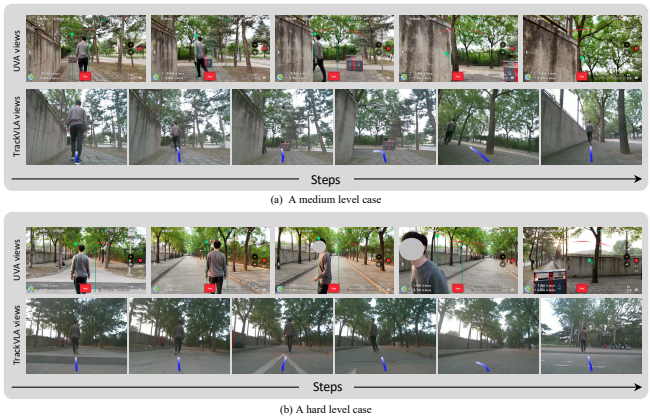
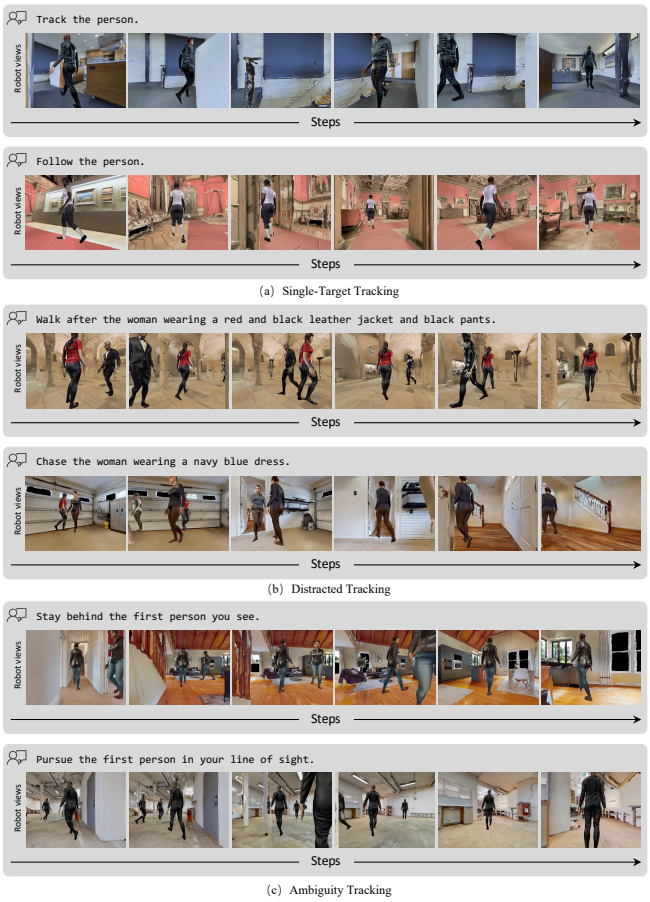


Fig. 9: **Visualization of the real-world experiments.** TrackVLA demonstrates robust tracking performance under challenging conditions such as occlusions and fast target motion, outperforming existing commercial tracking drones.

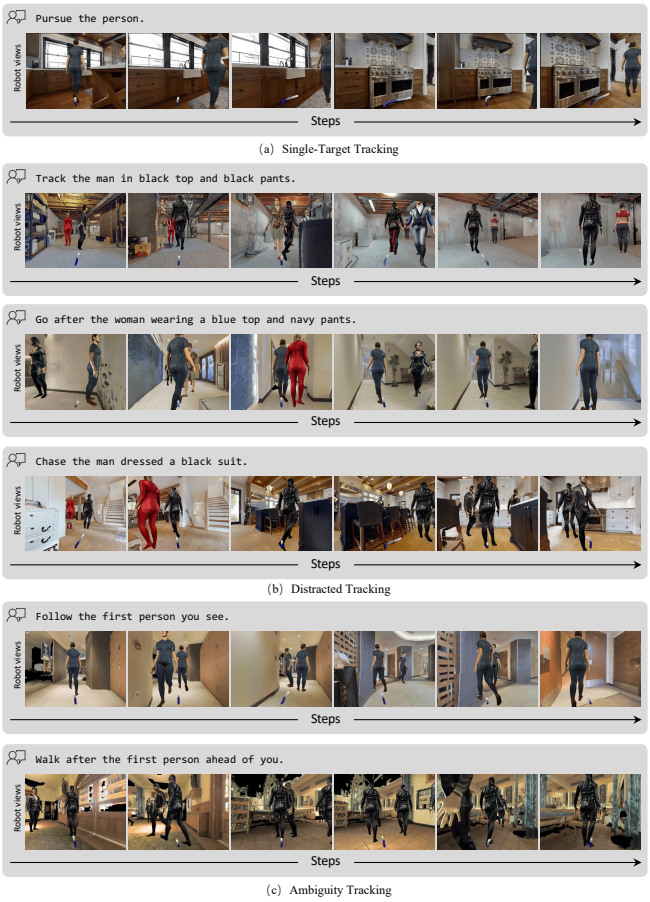


(a) Single-Target Tracking

(b) Distracted Tracking

(c) Ambiguity Tracking

Fig. 10: Visualization of the training set of EVT-Bench.

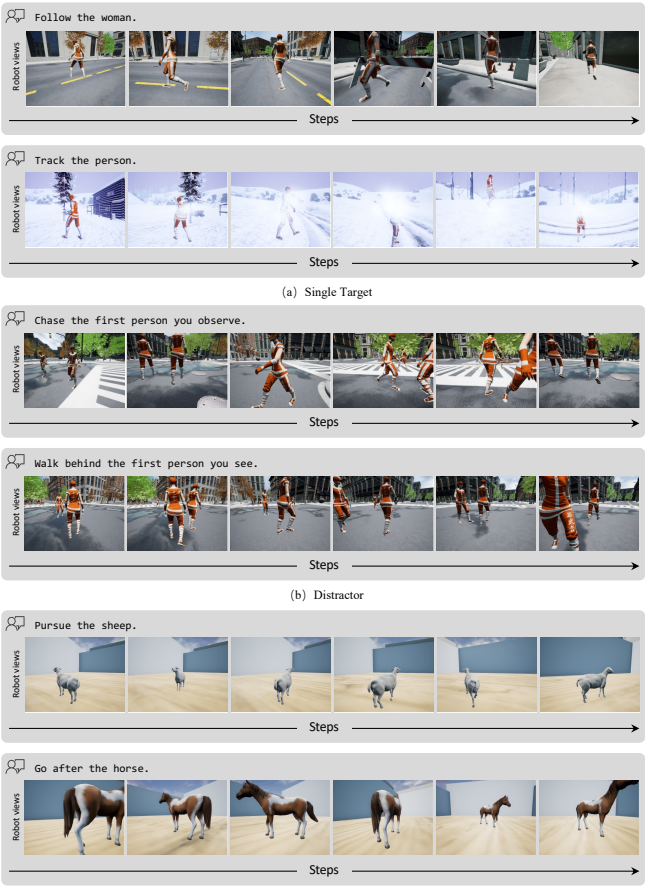


(a) Single-Target Tracking

(b) Distracted Tracking

(c) Ambiguity Tracking

Fig. 11: Visualization of TrackVLA on EVT-Bench.



(a) Single Target

(b) Distractor

(c) Unseen Objects

Fig. 12: Visualization of TrackVLA on Gym-UnrealCV.