

RSPT: Reconstruct Surroundings and Predict Trajectories for Generalizable Active Object Tracking

Fangwei Zhong^{*1, 2}, Xiao Bi^{*3}, Yudi Zhang⁴, Wei Zhang⁴, Yizhou Wang^{3, 5}

¹ Sch'l of Intelligence Science and Technology, Peking University

² Nat'l Key Lab. of GAI, Beijing Institute for General Artificial Intelligence (BIGAI)

³ Center on Frontiers of Computing Studies, Sch'l of Computer Science

Inst. for Artificial Intelligence, Peking University

⁴ Sch'l of Control Science and Engineering, Shandong University

⁵ Sch'l of Info. Eng., Zhengzhou University

{zfw, bixiao, yizhou, wang}@pku.edu.cn, reed_zyd@mail.sdu.edu.cn, davidzhang@sdu.edu.cn

Abstract

Active Object Tracking (AOT) aims to maintain a specific relation between the tracker and object(s) by autonomously controlling the motion system of a tracker given observations. AOT has wide-ranging applications, such as in mobile robots and autonomous driving. However, building a generalizable active tracker that works robustly across different scenarios remains a challenge, especially in unstructured environments with cluttered obstacles and diverse layouts. We argue that constructing a state representation capable of modeling the geometry structure of the surroundings and the dynamics of the target is crucial for achieving this goal. To address this challenge, we present RSPT, a framework that forms a structure-aware motion representation by Reconstructing the Surroundings and Predicting the target Trajectory. Additionally, we enhance the generalization of the policy network by training in an asymmetric dueling mechanism. We evaluate RSPT on various simulated scenarios and show that it outperforms existing methods in unseen environments, particularly those with complex obstacles and layouts. We also demonstrate the successful transfer of RSPT to real-world settings.

Introduction

Active object tracking (AOT) aims to follow a target object by autonomously controlling the motion system of an embodied agent. Specifically, the agent must perceive the movement of the target and its surroundings and subsequently adjust its posture to continuously position the target at the center of its view with an appropriate size. AOT has a vast array of applications, including drones (Ci et al. 2023), mobile robots (Wang et al. 2018), and autonomous driving (Jin and Han 2022).

Although recent years have witnessed remarkable progress in embodied AI (Chrisley 2003), it remains challenging for an agent to actively track a moving object in complex unstructured scenarios (Zhong et al. 2019a). In

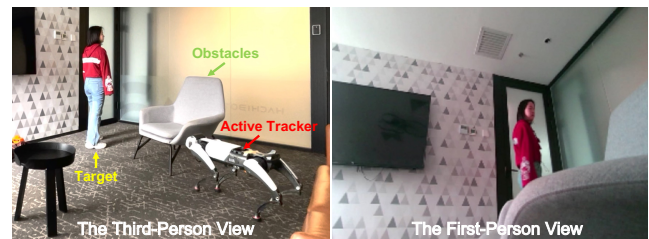


Figure 1: This exemplar case highlights the challenges of active object tracking, particularly in environments with cluttered obstacles. In this scenario, the target (a human) is occluded by obstacles, and the tracker (a robot) must find a way to follow the target without encountering occlusions.

these environments, obstacles are usually cluttered and arranged in a variety of ways. For example, in a family scene, different families have different room layouts, different furniture placements, and different appearances. The outdoor environment is more complex, *e.g.*, unpredictable obstacles may be encountered at any time. As shown in Figure 1, the tracker (robot) is required to find a path to follow the target (human) while avoiding occlusion and collision. In these environments, the performance of existing models, especially end-to-end networks (Luo et al. 2018; Zhong et al. 2019b), are dropped significantly.

The inadequate generalization of these models can be attributed to two primary factors: *visual perception* and *physical movement*. The diversity in appearances and layouts of obstacles results in various unseen situations for the tracker. Additionally, target occlusion by obstacles makes it invisible to the tracker, hindering data-driven methods. In terms of physical movement, obstacles may impede the tracker's path, necessitating route planning to bypass obstacles and re-localize the target.

To develop a generalizable tracker, it is essential to construct a structure-aware motion representation that meets three requirements: 1) abstraction of the scene's geometric structure to identify target objects, obstacles, and free space; 2) prediction of the target's future movement based on geo-

*These authors contributed equally.

metric structure; and 3) real-time computation of the overall pipeline.

In this paper, we propose a novel framework for generalizable active object tracking called **RSPT** (Reconstruct Surrounding and Predict Trajectory). RSPT consists of four modules: Target Localization, Structure Reconstruction, Structure-aware Trajectory Prediction, and a Motion Controller. The *Target Localization* estimates the 2D location of the target using an off-the-shelf video tracker and transforms it into 3D space with depth images. The *Structure Reconstruction* module builds a tracker-centric grid map of the environment in real-time using depth images and camera poses. The *Structure-aware Trajectory Prediction* module integrates historical relative trajectory data with the reconstructed map to predict future target movement. Finally, the Motion Controller, trained by Reinforcement Learning (RL) (Sutton and Barto 1998; Mnih et al. 2015), outputs actions to move the agent step-by-step using the constructed structure-aware motion representation. We adopt the asymmetric dueling mechanism (Zhong et al. 2019b,a) to improve tracker robustness in complex environments.

We demonstrate that the RSPT-based tracker outperforms previous AOT methods in terms of accumulated reward, episode length, and success rate in both simple and complex environments with cluttered obstacles and diverse layouts. Experimental results show that when obscured or blocked by obstacles, the tracker can predict the target’s movement and plan a path considering surrounding structure to follow the target. Even when the target leaves the field of view, the tracker can retrieve it and continue tracking using memory. Additionally, we deploy the tracker on a real-world robot to demonstrate its strong generalization capabilities both quantitatively and qualitatively.

The contributions of our work can be summarized as:

1. we introduce a structure-aware motion representation that combines the structure of surroundings and dynamic of the moving target in a grid map. This allows for improved tracking route planning in complex environments.
2. We present a practical framework that integrates off-the-shelf methods for structure reconstruction, video tracking, trajectory prediction and policy learning to achieve the desired representation.
3. We evaluate trackers in six unseen virtual environments and three real-world robot scenarios, demonstrating strong cross-domain generalization of our RSPT framework.

Related Work

Active Object Tracking (AOT). We can divide the AOT methods into two branches: one-stage methods and two-stage methods. Traditional methods perform in the two-stage manner (Kim et al. 2005; Hong et al. 2018), where the perception module provides a handcrafted state representation, *e.g.*, the 2D bounding box of the target (Ross et al. 2008; Mei and Ling 2009; Hu et al. 2012), then the controller actively adjusts the camera poses accordingly. Such a solution performs well in most simple cases, however, fails to handle complex cases, *e.g.*, occlusions. In recent years,

Deep Reinforcement Learning (DRL) is employed by some works (Luo et al. 2018; Zhang et al. 2018; Luo et al. 2020) to realize AOT tracker in an end-to-end manner. In the follow-up works (Zhong et al. 2019a; Devo, Dionigi, and Costante 2021; Dionigi et al. 2022; Zhong et al. 2021; Xi et al. 2021) demonstrated that multi-agent games can significantly improve tracker’s generalization on unseen target, occlusions or distractors. (Li et al. 2020) proposed a multi-camera collaboration solution to track the target in complex environments for PTZ camera networks. However, it is still difficult to deploy these methods in complex real-world environments, as the sim2real gap. In this work, we aim to build a generalizable tracker that performs well in unseen and complex real-world environments. We focus on state representation and introduce a novel structure-aware motion representation for the tracker. Additionally, we employ an asymmetric dueling mechanism in learning based on AD-VAT.

Structure Reconstruction. Map represents the geometry of a scene. Previous works (Zhang and Singh 2014, 2015; Zhong et al. 2018; Campos-Macías et al. 2021) have used the point clouds from LiDAR or RGB-D camera as input for map reconstruction. However, point clouds are high-dimensional and inefficient for downstream tasks such as policy learning. Autonomous navigation frameworks (Campos-Macías et al. 2021) have been proposed using depth sensors or visual SLAM (Campos et al. 2021; Qin, Li, and Shen 2018) in unknown environments, but these methods have limitations such as cumulative errors and high storage requirements. (Usenko et al. 2017) proposed a robot-centric 3D circular buffer to represent the local environment efficiently. In AOT, robots rely more on their surroundings than the global environment, making a local robot-centric map more suitable and efficient for tracking.

Trajectory Prediction. Human motion prediction is previously studied in computer vision (Zhao et al. 2019; Tang and Salakhutdinov 2019). (Lee et al. 2017; Nikhil and Morris. 2018) draw historical position on a grid map to take its sequential and spatial properties into consideration, resulting in increased computation operations. At present, the main method of historical position still takes it as sequential data, through Recurrent Neural Networks (RNN) (Medsker and Jain 2001) to extract the features. Recently, (Kim et al. 2017) proposed a method based on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) to predict the future trajectory of the target over the occupancy grid map. (Ridel et al. 2020) propose a model based on grid representations to forecast agent trajectories. They encode the scene and past trajectories using convolutional layers and generate trajectory forecasts using a Convolutional LSTM (ConvLSTM) (Xingjian et al. 2015). In AOT, the tracker is required to predict the target’s motion from a first-person view. To account for cluttered obstacles, we integrate scene structure and historical trajectory of the target to forecast the target’s trajectory distribution.

Method

The RSPT framework aims to create a structure-aware motion representation, denoted by Φ_t , using the RGB image I_t , the depth image D_t , and the camera pose of the

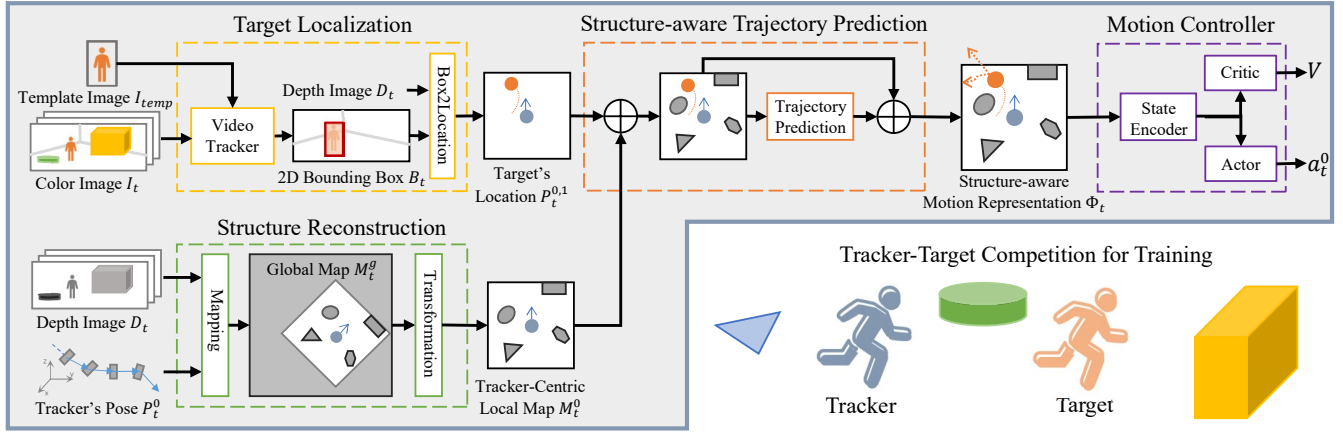


Figure 2: An overview of the RSPT framework for active object tracking. It forms a structure-aware motion representation by Reconstructing the Surroundings of the tracker and Predicting the Trajectory of the target. The tracker first localizes the target by a video tracker, and simultaneously constructs a local grid map with the depth image and camera pose, then predicts the future trajectory of the target in the map. Based on the representation, the controller learns the tracking policy via Reinforcement Learning (RL) under the asymmetric dueling mechanism (Zhong et al. 2019a).

tracker, P_t^0 . The framework consists of four main components, namely, *target localization*, *structure reconstruction*, *structure-aware trajectory prediction*, *motion controller*, as shown in Figure 2. By using this state representation, a generalizable tracker can be trained through reinforcement learning (RL) in an asymmetric dueling mechanism, as proposed in AD-VAT (Zhong et al. 2019b).

Target Localization

The most basic function of the tracker is to identify and localize the target. In this regard, we employ an off-the-shelf video tracker, denoted by VT , to estimate the 2D location of the target object in the image space. Specifically, we first crop the first RGB image I_0 according to the initial target bounding box B_0 , which serves as a template denoted by I_{temp} . Then, we use VT to detect the bounding box B_t of the target object in the subsequent RGB image I_t . To improve cross-domain generalization, we adopt DiMP (Bhat et al. 2019), which can online tune the network parameters to better fit the target appearance during testing. The overall formula can be expressed as follows:

$$B_t = VT(I_t, I_{temp}). \quad (1)$$

To model the historical trajectory of the target, it is necessary to transform the bounding box B_t in image space into the target's location $P_t^{0,1}$ in the camera coordinate system. Specifically, we obtain the central location (u, v) of the bounding box B_t and the corresponding depth value d in the depth image D_t . Next, we use the camera intrinsic parameters K to convert the (u, v, d) triplet into the target's 3D relative position $P_t^{0,1}$. The formula for this transformation is as follows:

$$P_t^{0,1} = CT(K, D_t, B_t). \quad (2)$$

The coordinate transforms CT is used to obtain the relative pose of the target with respect to the tracker. Here K , D_t and B_t represent the camera intrinsic parameters, the depth image, and the 2D bounding box of the target, respectively. The coordinates (x, y) represent the position $P_t^{0,1}$ of the target relative to the tracker, where x and y correspond to the lateral and longitudinal directions, respectively. At each time step, we set the tracker's coordinate as $(0, 0)$, and the target's coordinate at time step t is denoted by (x_t, y_t) .

To help the tracker in inferring the current location of the target, we define the visibility state of the target $Vis_t^1 = [P_t^{0,1}, P_{last}^1, C_{inv}]$. Specifically, $P_t^{0,1}$ represents the current position of the target relative to the tracker, P_{last}^1 represents the last observed position of the target, and C_{inv} denotes the number of consecutive time steps during which the target has not been observed by the tracker. In instances where the target is not visible, the video tracker outputs a "not found" flag and sets the relative position $P_t^{0,1}$ to all zeros. Notably, all position values are relative to the tracker.

Structure Reconstruction

The aim of the structure reconstruction module SR is to reconstruct the environment in real-time to represent the recent state of the tracker's surroundings. Visual SLAM systems (Campos et al. 2021) can perform this task, but their maps are not suitable for downstream tasks in real time. This is because processing 3D point clouds requires heavy computation, and sparse keypoint clouds provided by most SLAM systems lack necessary information for decision-making, such as the absence of points in a blank wall.

Motivated by this, we utilize a grid map (Fankhauser, Bloesch, and Hutter 2018) that is based on the depth image D_t and the estimated pose P_t^0 at each time step. To reduce computation costs, we recover point clouds within a certain height from the depth image and camera intrinsic parameters K . This approach may lead to a loss of map precision,

especially in scenarios with large variations in ground height and variable obstacle sizes. Nevertheless, we argue that this trade-off between accuracy and computational efficiency is acceptable. Empirical results demonstrate that such maps are generally suitable for downstream decision-making tasks. We then update the occupancy probability of each grid cell based on the density of the corresponding point cloud, allowing the grid map to reflect the probability of being occupied by obstacles, including dynamic ones like the target. Additionally, the 2D grid map is more amenable to real-time computation by convolutional neural networks, which is necessary for downstream tasks such as trajectory prediction and motion control. We combine obstacle information seen at different times with the tracker pose P_t^0 , and finally obtain the tracker-centric local map M_t^0 by transforming the grid-based map to the current pose coordinate. Note that the pose can be obtained from a localization system equipped in most robots or simulated by adding noise to a grounded pose in virtual environments. The overall formula is as follows:

$$M_t^0 = SR(D_t, K, P_t^0). \quad (3)$$

The local grid map is partitioned into fixed-size rectangular grids of equal size. To ensure a valid distance range of the RGB-D sensors and avoid accumulating errors, we limit the coordinate of the surrounding structure to within -10 to 10 meters. In our method, we employ a grid size of (40×40) , with each grid spanning 50cm in both length and width. This grid-based representation can be efficiently processed by a Neural Network for real-time decision making.

Structure-aware Trajectory Prediction

To predict the trajectory of the target on the map, we employ a trajectory predictor TP to estimate the target trajectory $\tau_{t+1:t+F}^1$. At each time step, we use the most recent H samples of the target's coordinates, $\tau_{t-H+1:t}^1 = [(x_{t-H+1}, y_{t-H+1}), \dots, (x_t, y_t)]$, to forecast the F subsequent coordinates, $\tau_{t+1:t+F}^1 = [(x_{t+1}, y_{t+1}), \dots, (x_{t+F}, y_{t+F})]$. We formulate this module as follows:

$$\tau_{t+1:t+F}^1 = TP(E_{map}(M_t^0), E_{traj}(\tau_{t-H+1:t}^1)). \quad (4)$$

We employ Convolutional Neural Networks (CNN) as a feature extractor E_{map} to capture the spatial surrounding structure. For the sequential information of the target's historical location $\tau_{t-H+1:t}^1$, we use Long Short-Term Memory (LSTM) as a feature extractor E_{traj} . In order to predict the target's trajectory, we compare the performance of simple regression and Mixture-Density Recurrent Network (Hu, Zhan, and Tomizuka 2018; Makansi et al. 2019) (MDN-RNN). Our experiments demonstrate that regression can only model the average distribution of the trajectories, whereas MDN-RNN can capture the distribution of the trajectories and generate multiple possible future trajectories. Specifically, for each future time step, MDN-RNN outputs the core weights, the mean and variance of the relative coordinate x and y in each GMM core, and concatenates them to form the target trajectory τ_t^1 . The final trajectory distribution

is then computed accordingly. Our PT module has two key advantages. Firstly, it conditions the trajectory prediction on the spatial distribution and size of obstacles in the environment, leading to more accurate trajectory forecasts in cluttered settings. Secondly, it estimates the likelihood of multiple potential paths, allowing the tracker to react more flexibly and rapidly to changes in the target's behavior, rather than assuming a regular movement pattern. We validate the effectiveness of our PT module by comparing it against a Kalman Filter, as detailed in Section .

Then, we construct a structure-aware motion representation that comprises the visibility state of the target, the tracker-centric local map, and the predicted trajectory, denoted as $\Phi_t = [Vis_t^1, E'_{map}(M_t^0), E'_{traj}(\tau_t^1)]$. Notably, the encoder E'_{map} and E'_{traj} employ the identical network architecture as E_{map} and E_{traj} , respectively, but possess independent network parameters for learning.

Motion Controller

We propose a neural network-based motion controller, denoted by $a_t = MC(\Phi_t)$, which takes the structure-aware motion representation Φ_t as input and outputs the action for the tracker at time step t . The motion controller is trained through reinforcement learning (Luo et al. 2020), where the objective is to maximize the expected cumulative reward over a finite horizon.

To enhance tracker robustness, we employ the asymmetric dueling mechanism (Zhong et al. 2019a) to automatically generate diverse target trajectories while training. Specifically, the target is an end-to-end network, and its input is the grounded location of all agents obtained in the simulation. This allows the target to identify weaknesses in the tracker's movements, thereby learning from successful escape paths. The reward structure is based on AD-VAT+ (Zhong et al. 2019a). Specifically, the reward for the tracker is defined as $r_0 = 1 - \Delta\rho - \Delta\theta$, where $\Delta\rho$ and $\Delta\theta$ denote the errors in relative distance and angle, respectively. As the target approaches the expected position, the tracker's reward increases. The target's reward is the negative of the tracker's reward written as $r_1 = -r_0$, incentivizing the target to evade the tracker. By training with an adversarial target, the robustness of the tracking policy is improved concurrently.

Experiments

In this section, we conduct experiments to address the following questions: 1) How does the performance of the proposed framework compare with that of existing methods? 2) To what extent does each module contribute to the overall improvement of the framework? 3) What is the primary bottleneck for active object tracking? 4) Can the proposed framework effectively handle noisy observations? 5) Can the framework be successfully deployed in real-world robotic applications? These experiments provide insights into the effectiveness and limitations of our approach, as well as its potential for practical applications.

The details of the experiments are introduced as follows.

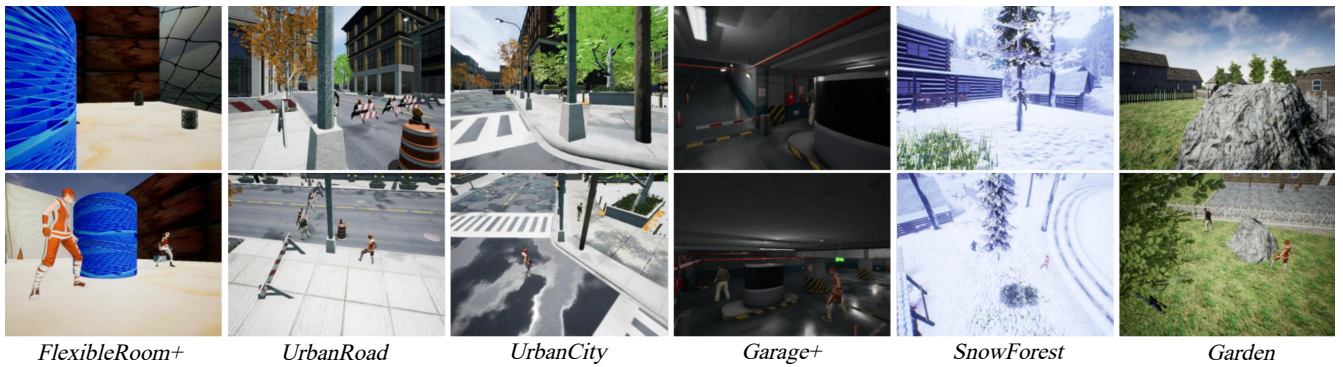


Figure 3: Examples of the environments for training (leftmost) and testing (others). The top row features first-person views of the trackers, while the bottom row displays third-person views

Environments

We employ six high-fidelity environments from AD-VAT+ (Zhong et al. 2019a) to evaluate our approach, as shown in Figure 3. These environments contain diverse obstacles with varying shapes and layouts among these environments. The *FlexibleRoom+* environment is used for training, where we randomize the layouts, shapes, and sizes of obstacles in each episode. The remaining six realistic environments (*UrbanCity*, *UrbanRoad*, *Garden*, *Garage*, *Garage+*, and *SnowForest*) are utilized to evaluate the generalization of the trackers in unseen environments. The action space is akin to (Zhong et al. 2019b), with seven discrete actions: *Move-forward*, *Turn-left*, *Turn-left-and-move-forward*, *Turn-right*, *Turn-right-and-move-forward*, *Stop*, and *Move-backward*.

Evaluation Metric

In our experiment, we set the maximum length of each episode to 500 time steps. Besides, a visible area is defined as a fan shape area in front of the tracker with a radius of 750cm and a range of 90 degrees. The target is considered lost if it moves outside of this visible area, and the episode terminates if the target remains lost for more than 5 seconds. To evaluate the tracking performance, each method runs over 50 episodes in each environment, and we report average results for three metrics: Accumulated Reward, Episode Length, and Success Rate. 1) **Accumulated Reward (AR)**. This metric, consistent with other RL tasks, measures the cumulative reward across an entire episode, and reflects both accuracy and robustness of the tracker. 2) **Episode Length (EL)**. We calculate the average episode length based on the defined episode termination condition, which reflects the long-term tracking performance. 3) **Success Rate (SR)**. An episode is considered successful if its length reaches the maximum of 500 steps. We calculate the rate of successful episodes across all episodes, which serves as a measure of model robustness. Higher values indicate greater robustness.

Comparison with Baselines

In the following, we compare our model with four two-stage methods and two end-to-end methods in six realistic unseen environments. Results are shown in Table 1.

Comparison with Two-stage Methods. In the two-stage methods, there are a visual perception module and a controller. For perception, we use an off-the-shelf video tracker (VT) or a semantic segmentation model (Seg) to encode the raw-pixel observation into a state representation, *e.g.*, the bounding box of target or the pixel-level segmentation mask. Specifically, we employ pretrained DiMP (Bhat et al. 2019) to track objects in videos and UNet + FCN from MMSegmentation¹ for semantic segmentation. For control, we build three kinds of control strategy, namely Rule, Planner, and RL. To be specific, the **Rule**-based controller is a PID-like controller, which outputs action based on the errors between the target location and the expectation. The **Planner** will reconstruct the surrounding map and uses the path planning algorithms, *e.g.*, A* (Hart, Nilsson, and Raphael 1968) to navigate the tracker to the expected distance next to the target. The **RL**-based method is trained via deep reinforcement learning to handle high-dimensional state representation.

First, we can see that RL-based controller achieved the best performance among the three VT-based method (*VT + Rule*, *VT + Planner*, *VT + RL*), showing the good tracking performance of the RL-based method. The *VT + Planner* adds path planning to achieve obstacle avoidance. However, it always has a delay, leading to the worst performance. Intuitively, when the tracker plans the path and starts walking according to optimal location in the current state, the target has left its previous position. This makes it difficult to maintain a specific distance and angle with the target. We also notice that *VT + RL* and *Seg + VT + RL* achieve comparable results, showing that additional semantic information about the scene is useless to the tracker.

Comparison with End-to-end Methods. We compare our proposed RSPT with three end-to-end methods (AD-VAT, AD-VAT+, TS), which also utilize RGB-D data as inputs. It is worthy to notice that the AD-VAT tracker was trained in the *FlexibleRoom* without any obstacles. Differ-

¹<https://github.com/open-mmlab/mms Segmentation>

Methods	<i>UrbanCity</i>	<i>UrbanRoad</i>	<i>Garage</i>	<i>Garage+</i>	<i>Garden</i>	<i>SnowForest</i>	Mean
VT + Rule	263/428/0.72	250/452/0.75	261/418/0.74	223/385/0.67	208/381/0.61	119/301/0.43	221/394/0.65
VT + Planner	215/476/0.84	200/ 464/0.84	145/445/0.70	162/421/0.66	173/ 450 /0.74	72/ 343/0.46	161/433/0.71
VT + RL	363/500/1.00	355 /490/0.94	284/427/0.70	239/393/0.56	234/394/0.56	152/295/0.38	271/416/0.69
Seg + VT + RL	303/437/0.83	311/469/0.91	283/425/0.65	243/416/0.63	205/403/0.67	232/364/0.59	263/419/0.71
AD-VAT	335/484/0.88	246/429/0.60	86/302/0.20	39/273/0.10	112/297/0.16	169/364/0.44	164/358/0.39
AD-VAT+	389 /497/0.94	326/471/0.80	267/439/0.60	166/366/0.40	108/277/0.16	182/365/0.44	240/402/0.55
TS	341/496/0.94	308/480/0.84	265/472/ 0.89	237/ 462 / 0.78	55/209/0.02	234/ 424 /0.63	240/424/0.68
RSPT (Ours)	341/ 500 / 1.00	346/ 500 / 1.00	314 / 480 /0.80	285 /445/0.73	283 /444/ 0.72	248 /410/ 0.80	302 / 463 / 0.84

Table 1: The quantitative results compared with the baselines, where the best results are shown in bold. Note that the top four methods are two-stage methods. VT and Seg represent two kinds of perception modules, *i.e.*, VT = Video Tracker, Seg = Semantic Segmentation. Rule, Plan, and RL represent three kinds of controllers. TS indicates an end-to-end training tracker with the teacher-student learning strategy introduced in (Zhong et al. 2021). Note that all the end-to-end methods (AD-VAT, AD-VAT+, and TS) use the RGB-D image as the input. The three numbers in each cell represent Accumulated Reward (AR), Episode Length (EL), and Success Rate (SR) respectively.

Methods	<i>UrbanCity</i>	<i>UrbanRoad</i>	<i>Garage</i>	<i>Garage+</i>	<i>Garden</i>	<i>SnowForest</i>	Mean
VisT	239/401/0.66	168/308/0.33	166/327/0.48	100/215/0.40	154/315/0.34	128/278/0.53	159/307/0.45
VisT + RS	313/ 500 / 1.00	305/ 500 / 1.00	284/477/0.76	249/431/0.69	247/438/0.70	171/365/0.60	261/451/0.79
VisT + PT	335/ 500 / 1.00	316/451/0.80	311/472/0.78	244/390/0.63	277/417/0.68	184/380/0.61	277/435/0.75
RSPT (MDN-RNN)	341 / 500 / 1.00	346 / 500 / 1.00	314 / 480 / 0.80	285 / 445 / 0.73	283 / 444 / 0.72	248 / 410 / 0.80	302 / 463 / 0.84
RSPT (KF)	329/ 500 / 1.00	318/ 500 / 1.00	284/459/0.70	262/439/0.71	251/440/0.68	219/387/0.65	277/454/0.79

Table 2: The quantitative results compared with the ablations, where the best results are highlighted in bold. VisT denotes the visibility state of the target, RS represents Reconstruct Structure, PT indicates Predict Trajectory, and KF is a Kalman Filter that offers similar functionality to our PT module. Each cell displays three numbers, which represent the Accumulated Reward (AR), Episode Length (EL), and Success Rate (SR), respectively.

ently, **AD-VAT+** is an extension of AD-VAT and aims to handle more complex environments with obstacles via a two-stage learning strategy. Besides, we construct a variant of (Zhong et al. 2021), namely **TS**, we employ the teacher-student learning framework to train the end-to-end RGB-D tracker. Unlike (Zhong et al. 2021), the teacher takes grounded agent-centric grid maps as input for obstacle avoidance instead of relative agent poses. To ensure a fair comparison, we avoid introducing the distracting player in any of the training environments.

The results, as shown in Table 1, demonstrate that our method has a higher reward outperform the best end-to-end baseline by more than 100 points in *Garden*. Our success rate is also significantly higher than baselines in the *Garden* and *SnowForest*. The results indicates that it is challenging for end-to-end trackers to extract enough information about the non-current field of view by implicitly encoding raw-pixel observations, leading to suboptimal decision-making. In contrast, using our structure-aware motion representation allows the tracker to develop a better understanding of the surrounding environment and the targets’ movements, which is helpful to track in complex environments.

Ablation Study

We design some ablation versions of RSPT, namely VisT (Visibility of Target), VisT+RS (Reconstruct Structure), VisT+PT (Predict Trajectory), and RSPT (KF). The results are shown in Table 2.

VisT serves as the foundational component of the state representation, with additional RS, PT, and KF modules

added to create three other methods. Specially, in RS + KF, we replaces PT module in VisT + PT with an off-the-shelf Kalman Filter ² as our trajectory prediction module. This linear KF filter is used to forecast the future state of a dynamic process using a collection of (low-dimensional) observations.

Compared VisT with the VT + RL results in Table 1, VT + RL achieved better performance, indicating that only using the location of the visible target is helpless to improve the generalization of tracker. Our results indicate that RS enables the tracker to plan an optimal tracking route around obstacles by leveraging structural information. Meanwhile, PT enables the tracker to adjust its trajectory in anticipation of the target’s predicted motion. These improvements highlight the importance of incorporating both structural information and motion prediction in active object tracking. The performance gap observed between the two ablations (VisT + RS and VisT + PT) and our RSPT (MDN-RNN) demonstrates the effectiveness of the missing components. The performance gap between the two choices of PT module (MDN-RNN vs. KF) underscores the importance of accounting for surrounding obstacles and considering multiple possibilities for the target trajectory in trajectory prediction. Our results suggest that the MDN-RNN model’s ability to model the target’s motion uncertainty and leverage structural information leads to improved trajectory predictions and overall tracking performance compared to the simpler KF module. These findings demonstrate the value of incor-

²<https://github.com/zziz/kalman-filter>

	<i>UrbanCity</i>	<i>UrbanRoad</i>	<i>Garage</i>	<i>Garage+</i>	<i>Garden</i>	<i>SnowForest</i>	Mean
Object Mask	302/494/0.98	353/499/0.94	177/378/0.52	232/420/0.64	162/389/0.38	37/264/0.01	210/407/0.57
Target Mask	407/500/1.00	392/494/0.96	359/474/0.84	218/444/0.66	377/497/0.98	305/461/0.83	343/478/0.87
RSPT with GT	375/ 500/1.00	356/ 500/1.00	347/ 493/0.90	350/479/0.84	385/500/1.00	384/500/1.00	366/492/0.93
RSPT (Ours)	341/ 500/1.00	346/ 500/1.00	314/480/0.80	285/445/0.73	283/444/0.72	248/410/0.80	302/463/0.84

Table 3: The quantitative results with different grounded state. The top two methods take Object Mask, and Target Mask as input, separately, while the RSPT with GT method replace the estimated pose in RSPT with the grounded pose. Note that the RSPT with GT methods provides an upper bound of our RSPT methods. The three numbers in each cell represent Accumulated Reward (AR), Episode Length (EL), and Success Rate (SR) respectively.

Noisy Input	<i>UrbanCity</i>	<i>UrbanRoad</i>	<i>Garage</i>	<i>Garage+</i>	<i>Garden</i>	<i>SnowForest</i>	Mean
Clean Input	341/500/1.00	346/500/1.00	314/480/0.80	285/445/0.73	283/444/0.72	248/410/0.80	302/463/0.84
Depth	350/500/1.00	331/473/0.82	281/445/0.73	264/411/0.68	259/417/0.69	204/375/0.62	281/436/0.75
Pose	339/500/1.00	347/500/1.00	305/477/0.77	262/429/0.64	266/436/0.70	229/404/0.78	291/457/0.81
Depth + Pose	336/500/1.00	328/466/0.81	244/407/0.62	265/423/0.66	254/402/0.64	198/368/0.57	270/427/0.71

Table 4: Evaluating RSPT with different noisy inputs. The four methods take grounded information, noise depth + grounded pose, grounded depth + noise pose, and noise depth + noise pose as input, separately. The three numbers in each cell represent Accumulated Reward (AR), Episode Length (EL), and Success Rate (SR) respectively.

porating advanced motion models and structure-aware trajectory prediction.

Tracking with Grounded States

To identify the bottleneck for AOT, we devised two state representations utilizing different grounded states: semantic segmentation masks and depth images. The first representation, **Object Mask**, employs instance-level object segmentation masks and depth images as inputs. The second representation, **Target Mask**, makes use of binary masks of the target and background, as well as depth images as inputs. Additionally, we constructed **RSPT with GT** by replacing the estimated poses in RSPT (which are based on video tracker and depth image) with ground truth poses. This method serves to verify the upper-bound of the structure-aware motion representation-based approach without being impacted by the accuracy of the video tracker.

Shown as Table 3, the Target Mask model outperforms the Object Mask model due to the binary mask filtering out irrelevant background semantics. This allows the model to focus more on the target and the structure information, leading to improved performance. Despite the Target Mask model’s improved performance, it still exhibits significant shortcomings in dealing with various unseen environments due to a lack of awareness of environmental structure and target movement. In contrast, utilizing the proposed RSPT method for constructing the tracker achieved a 100% success rate in four out of six diverse testing environments, demonstrating that this approach can significantly enhance robustness in complex environments.

Robustness Analysis

To evaluate the robustness of the trackers, we introduce noise to the input and conduct experiments. The video tracker is trained on real data, bridging the gap between simulation and reality in RGB processing. However, the depth image captured by a real RGB-D sensor is often noisy, and

the mapping method is frequently affected by cumulative errors of the pose, making noise on camera poses a significant concern. To add noise to the depth image, we use the random-shift/laterally-corruption cooperation from the ICL-NUIM dataset (Handa et al. 2014). Additionally, we apply Gaussian noise with a mean of 0 and a variance of 0.5 meters to the pose to simulate noisy pose.

As shown in Table 4, our method is not greatly affected by pose errors by constructing tracker-centric surrounding maps. The noise on depth causes more drop in the performance, but our method with noisy input remains achieved higher scores than end-to-end trackers with clear depth.

Real-world Deployment

To verify the practical value of the proposed RSPT tracker, trained in *FlexibleRoom+* with 1m expected distances, we conduct experiments on a physical robot in real-world settings. The tracker’s performance is tested in three situations: Moving Forward, Moving Backward, and Tracking with Visual Occluding. *Moving Forward* involves the target winding around the obstacle or S-type line, testing the tracker’s ability in obstacle avoidance and target movement prediction. *Moving Backward* tests the tracker’s ability to remember obstacles outside the field of view as the target moves along the reverse direction of the mobile robots into the edge of the barrier. *Tracking with Visual Occluding* tests the tracker’s ability to handle visual occlusion as the target hides behind the obstacle.

In terms of quantification, we mainly consider the success rate, distance error, and direction error as (Luo et al. 2020). We collected 79 minutes of real-world tracking sequences, a total of 47k steps, to evaluate the model. For each frame, we calculate the three indicators using the bounding box, and compute the average value over all frames for each situation. A higher success rate indicates a more robust tracker, while smaller distance error and direction error correspond to higher tracking accuracy.

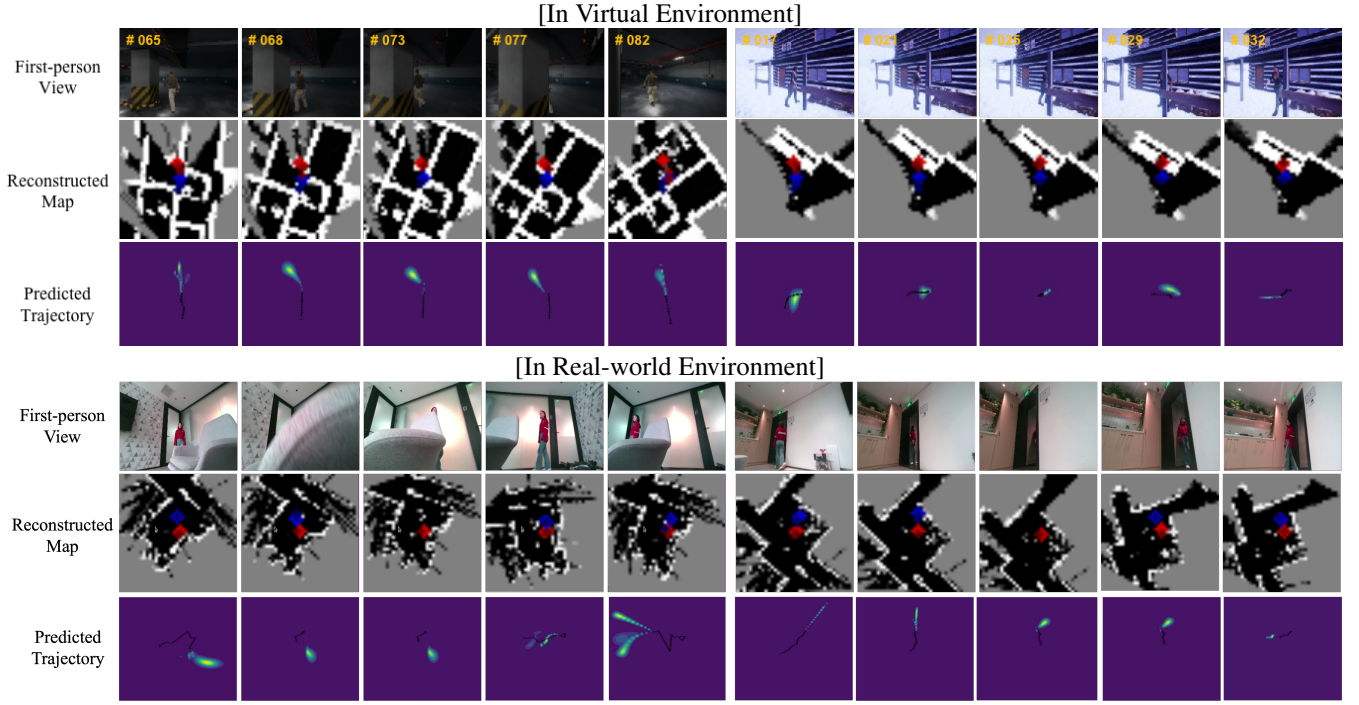


Figure 4: Exemplar sequences of RSPT in different environments. The reconstructed map depicts the target as a red dot, the tracker as a blue dot, obstacles as white, free space as black, and the unexplored area as gray. Meanwhile, the predicted trajectory displays the historical trajectory via a black line and the future trajectory distribution via a green area. The latter reflects the probability density function, with brighter colors indicating higher probabilities.

Situation	Method	SR	Distance Error	Direction Error
MF	VT + Planner	0.90	0.21 ± 0.05	0.13 ± 0.07
	RSPT	0.97	0.14 ± 0.03	0.08 ± 0.06
MB	VT + Planner	0.83	0.29 ± 0.08	0.20 ± 0.09
	RSPT	0.91	0.25 ± 0.09	0.11 ± 0.05
VO	VT + Planner	0.68	0.32 ± 0.10	0.37 ± 0.12
	RSPT	0.95	0.17 ± 0.06	0.14 ± 0.08

Table 5: We present quantitative results in three real-world situations: Moving Forward (MF), Moving Backward (MB), and Visual Occluding (VO). The three numbers in each cell represent Accumulated Reward (AR), Episode Length (EL), and Success Rate (SR) respectively. We compare our results with the VT + Planner baseline and highlight the best results in bold.

The quantitative results are shown in Table 5. *VT + Planner* represents the traditional two-stage method. As the sim2real gap, all the end-to-end methods fail to deploy in our robot. Our proposed RSPT tracker outperforms *VT + Planner* in terms of success rate, distance error, and direction error. Specifically, the RSPT tracker achieved a higher success rate in the Moving Forward (MF) situation, which is the most frequently occurring situation in the training data. In addition, even in the Moving Backward (MB) situation with limited space, our approach maintains robust tracking performance. Moreover, our method demonstrates better performance than the traditional method in the Visual Occlud-

ing (VO) situation where the 2D tracker is ineffective. This is due to the ability of the RSPT method to utilize the predicted state of the target for control, which enables normal tracking to be maintained.

Exemplar Cases

For a better understanding of the workflow of the RSPT tracker, we further provide demo sequences in Figure 4, including simulation and reality scenarios.

Figure 4 shows that the RSPT tracker can predict the target’s position using map and historical trajectory information when visual occlusion occurs, enabling continuous tracking. In cases where the target is hiding in a corner, the tracker leverages map information to predict the target’s intention and remains stationary, awaiting the target’s emergence. However, the future trajectory proposed by the PT module appears suboptimal in some cases, as it must adapt to diverse learning targets with varying walking strategies. As future work, we plan to classify the target and predict its trajectory based on the pattern of movement.

Conclusion

In the field of active object tracking, we assert that accurate prediction of target trajectory and awareness of surrounding obstacles are critical for successful tracking, particularly in environments with clustered obstacles and diverse layouts. To address these challenges, we propose a structure-aware

motion representation to develop a generalizable RGB-D tracker. Our approach includes key modules such as target localization, structure reconstruction, structure-aware trajectory prediction, and motion controller. Through experimental results on a range of realistic virtual environments, we demonstrate that our approach exhibits superior generalization performance. Additionally, we confirm the effectiveness of our approach in real-world scenarios by deploying it on a quadruped robot, which shows promising sim-to-real generalization capabilities.

Our work opens up several interesting directions for future research. Firstly, each module of the RSPT tracker can be further developed to address specific challenges. For example, in the structure reconstruction module, it would be important to efficiently build 3D maps for uneven terrain. Additionally, the tracker should be designed to avoid dynamic obstacles such as distractors without colliding with other obstacles. Secondly, for the video tracker, a promising direction is to incorporate the structure representation with image features to improve robustness to distractors and occlusion. Thirdly, for trajectory prediction, we consider adopting the machine theory of mind approach to predict the target's intention (Wang et al. 2022) for more precise trajectory prediction. Furthermore, it would be valuable to explore ways to make the modules mutually beneficial. Lastly, our framework could be extended to more challenging multi-agent settings such as team formation (Jin et al. 2022), target coverage control (Xu, Zhong, and Wang 2020; Pan et al. 2022), 3D human pose estimation (Ci et al. 2023).

Acknowledgements

This work was supported by MOST-2022ZD0114900, China National Post-doctoral Program for Innovative Talents (Grant No. BX2021008), NSFC-62061136001, and Qualcomm University Research Grant.

References

- Bhat, G.; Danelljan, M.; Gool, L. V.; and Timofte, R. 2019. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 6182–6191.
- Campos, C.; Elvira, R.; Rodríguez, J. J. G.; Montiel, J. M.; and Tardós, J. D. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*.
- Campos-Macías, L.; Aldana-López, R.; de la Guardia, R.; Parra-Vilchis, J. I.; and Gómez-Gutiérrez, D. 2021. Autonomous navigation of MAVs in unknown cluttered environments. *Journal of Field Robotics*, 38(2): 307–326.
- Chrisley, R. 2003. Embodied artificial intelligence. *Artificial Intelligence*, 149(1): 131–150.
- Ci, H.; Liu, M.; Pan, X.; Zhong, F.; and Wang, Y. 2023. Proactive Multi-Camera Collaboration for 3D Human Pose Estimation. In *The Eleventh International Conference on Learning Representations*.
- Devo, A.; Dionigi, A.; and Costante, G. 2021. Enhancing continuous control of mobile robots for end-to-end visual active tracking. *Robotics and Autonomous Systems*, 142: 103799.
- Dionigi, A.; Devo, A.; Guiducci, L.; and Costante, G. 2022. E-VAT: An Asymmetric End-to-End Approach to Visual Active Exploration and Tracking. *IEEE Robotics and Automation Letters*, 7(2): 4259–4266.
- Fankhauser, P.; Bloesch, M.; and Hutter, M. 2018. Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization. *IEEE Robotics and Automation Letters*, 3(4): 3019–3026.
- Handa, A.; Whelan, T.; McDonald, J.; and Davison, A. J. 2014. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *2014 IEEE international conference on Robotics and automation (ICRA)*, 1524–1531. IEEE.
- Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*, 9(8): 1735–1780.
- Hong, Z.-W.; Yu-Ming, C.; Su, S.-Y.; Shann, T.-Y.; Chang, Y.-H.; Yang, H.-K.; Ho, B. H.-L.; Tu, C.-C.; Chang, Y.-C.; Hsiao, T.-C.; et al. 2018. Virtual-to-real: Learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*.
- Hu, W.; Li, X.; Luo, W.; Zhang, X.; Maybank, S.; and Zhang, Z. 2012. Single and multiple object tracking using log-Euclidean Riemannian subspace and block-division appearance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12): 2420–2440.
- Hu, Y.; Zhan, W.; and Tomizuka, M. 2018. Probabilistic Prediction of Vehicle Semantic Intention and Motion. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 307–313.
- Jin, K.; and Han, X. 2022. Conquering Ghosts: Relation Learning for Information Reliability Representation and End-to-End Robust Navigation. *arXiv preprint arXiv:2203.09952*.
- Jin, K.; Wang, J.; Wang, H.; Liang, X.; Guo, Y.; Wang, M.; and Yi, H. 2022. Soft formation control for unmanned surface vehicles under environmental disturbance using multi-task reinforcement learning. *Ocean Engineering*, 260: 112035.
- Kim, B.; Kang, C. M.; Kim, J.; Lee, S. H.; Chung, C. C.; and Choi, J. W. 2017. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 399–404. IEEE.
- Kim, K. K.; Cho, S. H.; Kim, H. J.; and Lee, J. Y. 2005. Detecting and tracking moving object using an active camera. In *International Conference on Advanced Communication Technology*, volume 2, 817–820.
- Lee, N.; Choi, W.; Vernaza, P.; Choy, C. B.; Torr, P. H.; and Chandraker, M. 2017. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 336–345.

- Li, J.; Xu, J.; Zhong, F.; Kong, X.; Qiao, Y.; and Wang, Y. 2020. Pose-Assisted Multi-Camera Collaboration for Active Object Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 759–766.
- Luo, W.; Sun, P.; Zhong, F.; Liu, W.; Zhang, T.; and Wang, Y. 2018. End-to-end active object tracking via reinforcement learning. In *International conference on machine learning*, 3286–3295. PMLR.
- Luo, W.; Sun, P.; Zhong, F.; Liu, W.; Zhang, T.; and Wang, Y. 2020. 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.
- Makansi, O.; Ilg, E.; Cicek, O.; and Brox, T. 2019. Overcoming Limitations of Mixture Density Networks: A Sampling and Fitting Framework for Multimodal Future Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Medsker, L. R.; and Jain, L. 2001. Recurrent neural networks. *Design and Applications*, 5: 64–67.
- Mei, X.; and Ling, H. 2009. Robust visual tracking using L1 minimization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1436–1443.
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.
- Nikhil, N.; and Morris, B. T. 2018. Convolutional neural network for trajectory prediction. *IEEE Conference on Computer Vision and Pattern Recognition*.
- Pan, X.; Liu, M.; Zhong, F.; Yang, Y.; Zhu, S.-C.; and Wang, Y. 2022. MATE: Benchmarking Multi-Agent Reinforcement Learning in Distributed Target Coverage Control. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Qin, T.; Li, P.; and Shen, S. 2018. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4): 1004–1020.
- Ridel, D.; Deo, N.; Wolf, D.; and Trivedi, M. 2020. Scene compliant trajectory forecast with agent-centric spatio-temporal grids. *IEEE Robotics and Automation Letters*, 5(2): 2816–2823.
- Ross, D. A.; Lim, J.; Lin, R.-S.; and Yang, M.-H. 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3): 125–141.
- Sutton, R. S.; and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st edition. ISBN 0262193981.
- Tang, C.; and Salakhutdinov, R. R. 2019. Multiple futures prediction. *Advances in Neural Information Processing Systems*, 32.
- Usenko, V.; Von Stumberg, L.; Pangercic, A.; and Cremers, D. 2017. Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 215–222. IEEE.
- Wang, M.; Liu, Y.; Su, D.; Liao, Y.; Shi, L.; Xu, J.; and Miro, J. V. 2018. Accurate and real-time 3-D tracking for the following robots by fusing vision and ultrasonic information. *IEEE/ASME Transactions On Mechatronics*, 23(3): 997–1006.
- Wang, Y.; Zhong, F.; Xu, J.; and Wang, Y. 2022. ToM2C: Target-oriented Multi-agent Communication and Cooperation with Theory of Mind.
- Xi, M.; Zhou, Y.; Chen, Z.; Zhou, W.; and Li, H. 2021. Anti-distractor active object tracking in 3d environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6): 3697–3707.
- Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 802–810.
- Xu, J.; Zhong, F.; and Wang, Y. 2020. Learning Multi-Agent Coordination for Enhancing Target Coverage in Directional Sensor Networks. In *Advances in Neural Information Processing Systems*, volume 33, 10053–10064.
- Zhang, J.; and Singh, S. 2014. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems*.
- Zhang, J.; and Singh, S. 2015. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2174–2181. IEEE.
- Zhang, W.; Song, K.; Rong, X.; and Li, Y. 2018. Coarse-to-fine UAV target tracking with deep reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 16(4): 1522–1530.
- Zhao, T.; Xu, Y.; Monfort, M.; Choi, W.; Baker, C.; Zhao, Y.; Wang, Y.; and Wu, Y. N. 2019. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12126–12134.
- Zhong, F.; Sun, P.; Luo, W.; Yan, T.; and Wang, Y. 2019a. 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.
- Zhong, F.; Sun, P.; Luo, W.; Yan, T.; and Wang, Y. 2019b. AD-VAT: An Asymmetric Dueling mechanism for learning Visual Active Tracking. In *International Conference on Learning Representations*.
- Zhong, F.; Sun, P.; Luo, W.; Yan, T.; and Wang, Y. 2021. Towards distraction-robust active visual tracking. In *International Conference on Machine Learning*, 12782–12792. PMLR.
- Zhong, F.; Wang, S.; Zhang, Z.; and Wang, Y. 2018. Detect-SLAM: Making object detection and SLAM mutually beneficial. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1001–1010. IEEE.