KURL: A Knowledge-Guided Reinforcement Learning Model for Active Object Tracking

Xin Liu liuxin_lx@nudt.edu.cn
Jie Tan* j.tanjie@outlook.com
Xiaoguang Ren* rxg_nudt@126.com
Weiya Ren weiyren.phd@gmail.com
Huadong Dai hddai@vip.163.com
Intelligent Game and Decision Lab (IGDL), Beijing, China

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Recent studies have shown that active object tracking algorithms based on deep reinforcement learning have the difficulty of model training while achieving favorable tracking outcomes. In addition, current active object tracking methods are not suitable for air-toground object tracking scenarios in high-altitude environments, such as air search and rescue. Therefore, we proposed a Knowledge-gUided Reinforcement learning (KURL) model for active object tracking, which includes two embedded knowledge-guided models (i.e., the state recognition model and the world model), together with a reinforcement learning module. The state recognition model utilizes the correlation between the observed states and image quality (as measured by object recognition probability) as prior knowledge to guide reinforcement learning algorithm to improve the observed image quality. The reinforcement learning module actively controls the Pan-Tilt-Zoom (PTZ) camera to achieve stable tracking. Additionally, a world model is proposed to replace the traditional Unreal Engine (UE) simulator for model training, which significantly enhancing the training efficiency (about ten times). The results indicate that the KURL model can significantly enhance the image quality, stability and robustness of tracking, compared with other methods in similar tasks.

Keywords: Active Object Tracking; Reinforcement Learning; Knowledge-guided; PTZ Cameras; Automatically Control.

1. Introduction

Active object tracking, a method that aims to autonomously adjust the camera (e.g., position and attitude) to retain detailed information about the trajectory of moving objects (Tiritiris et al., 2021), has become a research hotspot in target tracking with the advancement of machine vision (Luo et al., 2018). Regarding the adaptable platform for active object tracking, unmanned aerial vehicles (UAVs) win out due to their portability, flexibility, and high maneuverability. However, most studies have generally used multi-rotor UAVs to accomplish target tracking in low-altitude environments (Mittal et al., 2020), while there are few active object tracking methods for high-altitude environments using fixed-wing UAVs.

∗. Corresponding author

Compared with low-altitude environments, high-altitude environments have farther observation distances, resulting in lower image quality and difficulty in extracting useful information from the obtained image. In addition, atmospheric disturbances and complex ground environments can interfere with the tracking process and thus cause object loss, which poses a challenge to the performance of active object tracking methods. Although the combination of deep reinforcement learning and active object tracking methods has shown some initial results in improving tracking performance (Luo et al., 2018, 2019), certain issues such as control difficulty (Li et al., 2018) and model training difficulty (Zhao et al., 2021) should be acknowledged. Some studies have shown that introducing knowledge to active object tracking methods provide solutions to the above-mentioned challenges, i.e., it can not only optimize the tracking performance (Ma et al., 2021), but also improve the training efficiency (Wang et al., 2021).

Therefore, this paper proposes a novel Knowledge-gUided Reinforcement Learning (KURL) model to accomplish an air-to-ground (i.e., high-altitude UAV tracks a vehicle on the ground) object tracking task. Specifically, the KURL model includes two embedded knowledge-guided models, namely the state recognition model and the world model, together with a reinforcement learning module using an improved proximal policy optimization algorithm. The state recognition model based on the correlation between observation states and the quality of the observed images is established as prior knowledge, which guides the tracking process to improve the image quality. Moreover, the world model based on the environment abstraction is proposed as knowledge to replace the UE simulator, which provides crucial parameters involved in the simulation environment and then accelerates the training process of reinforcement learning models.

The results have shown that the KURL model demonstrates better stability than previous active object tracking methods in similar tasks across all the proposed scenarios with various vehicle motions and disturbance modes. Additionally, the KURL model displays significant robustness, especially after object loss, enabling active control of the PTZ camera to locate the lost target and resume tracking in time. In comparison, other state-of-the-art methods exhibit limited ability to recover target tracking, as the chances of the target reappearing in view are lower even after a temporary loss. Moreover, the KURL model can automatically adjust the magnification through knowledge guidance during tracking, consequently improving the quality of the observed images. In summary, our main contributions include the following:

- A new knowledge-guided reinforcement learning model is proposed for active object tracking in high-altitude environments, which outperforms the state-of-the-art methods in terms of tracking stability and robustness in all proposed scenarios, especially in the case of target loss (due to disturbances).
- A novel reward function incorporating both knowledge (i.e., state recognition model) and visual distance guidance is proposed. The contained knowledge can guide the reinforcement learning algorithm to tune the focus of PTZ camera to track the object and improve the input image quality.
- A memory-enabled actor-critic neural network is designed for active object tracking, while the training strategy of the PPO algorithm is also optimized.
- Introducing the world model as knowledge significantly improves the training speed (i.e., about 10 times that of the UE simulator).

2. Related Work

Active object tracking locks the object by autonomously adjusting the position and attitude of the camera through visual observation (Xi et al., 2021), which has been applied to a range of platforms, including PTZ cameras (Li et al., 2020), vehicles (Devo et al., 2021), and UAVs (Moon et al., 2021). For instance, Kyrkou (2021) proposed a real-time and lightweight active object tracking network C^{3}Net for roadside monitoring. Zhang et al. (2022) implemented an end-to-end tracking method for UAVs by introducing GRU into the reinforcement learning network. However, the above-mentioned studies are not suitable for air-to-ground tracking tasks in high-altitude environments due to the relatively close distance between the tracker and the object. To investigate the robustness of the tracking methods, researchers have introduced disturbance factors, including similar objects (Xi et al., 2021), occlusion (Cui et al., 2021; Zhong et al., 2021), and obstacles (Luo et al., 2021), to the training process. However, they do not consider the disturbance of vibration-induced tracker during tracking and the re-tracking after target loss.

To improve training efficiency, researchers have introduced some knowledge such as transfer learning and imitation learning directly into the models. For example, Li et al. (2021) accelerated neural network convergence by sharing model parameters learned from the source domain task with a new model in the target domain task. Zhong et al. (2021) formed a two-stage teacher-student learning strategy by transferring meta-policy knowledge to active visual trackers, which avoided multiple attempts and task exploration and improved training efficiency.

Another common method of knowledge introduction is to combine PID methods with reinforcement learning to form a hierarchical control framework, to improve training efficiency and control stability (Li et al., 2018). For example, Zhao et al. (2021) achieved end-to-end active target tracking by introducing PID methods into deep reinforcement learning to combine a high-level controller with a low-level controller in a hierarchical active tracking control framework. Ma et al. (2021) achieved trajectory tracking of underwater gliders by combining an onboard PID controller with the DDPG algorithm. Wang et al. (2021) used the PID method as a supervisory controller to guide policy network optimization, achieving efficient model training that outperformed the PID method in tracking performance.

The findings of the above-mentioned studies indicate that introducing knowledge into active object tracking methods can not only improve training efficiency, but also enhance tracking performance. However, although these methods achieve good tracking results, there are problems such as vibration, object loss, and low image quality in high-altitude object tracking scenarios, which limit the application of the existing used knowledge. Thus, we propose a knowledge-guided active object tracking method that is well-suited to high-altitude environments.

3. Approach

3.1. Overview

In this paper, we propose a knowledge-guided reinforcement learning (KURL) model to address the high-altitude active object tracking task. The KURL model (as shown in Figure 1) includes two embedded knowledge-guided models, namely the state recognition model and the world model, together with a reinforcement learning module using an improved proximal policy optimization algorithm (Schulman et al., 2017). A brief description of the knowledge-guided models is given below.

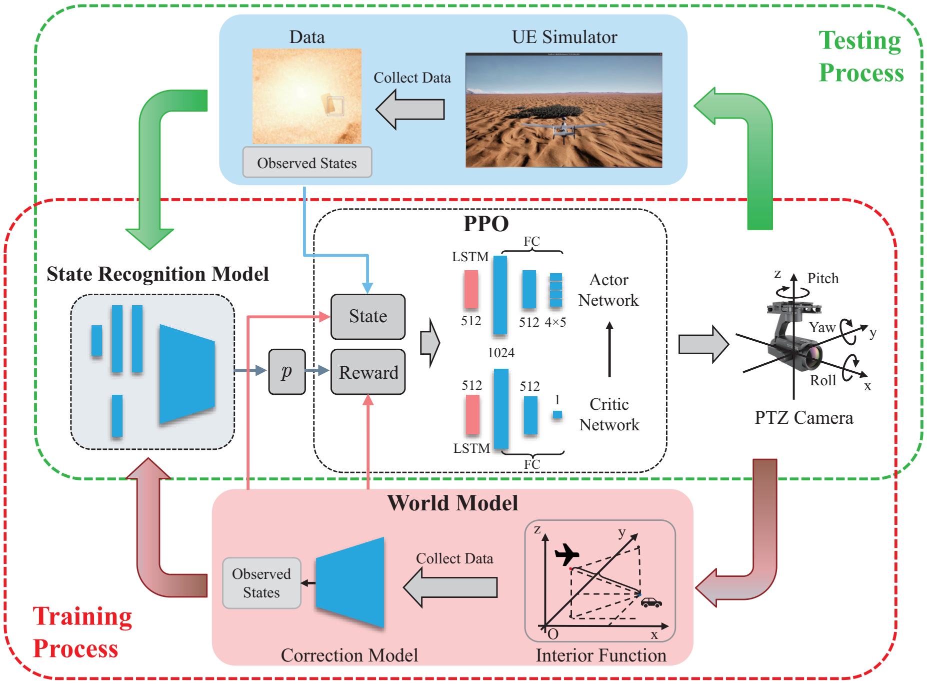

Figure 1: The overall framework.

The state recognition model establishes a relationship between the image quality (as measured by object recognition probability) and the observed camera states through supervised learning. Introducing this model as prior knowledge into reward shaping can guide the motion of PTZ cameras to further improve the quality of observed images, obtain additional information, and enhance the object tracking performance. Meanwhile, it can also avoid the vast computational burden caused by direct image processing with reinforcement learning.

The introduction of the world model as knowledge solves the problems of high resource consumption and slow training speed encountered in the training of traditional simulator models based on UE engines. The interior function of the world model is responsible for generating the necessary parameters required for training, while the correction model compensates for the control errors between the generated parameters and the UE simulator. The reinforcement learning model trained on the world model can be directly tested and used in the UE simulator, greatly improving the training efficiency.

3.2. State Recognition Model

Figure 2 illustrates the structure of the state recognition model. The model takes the observed states as the input and the object recognition probability measuring the image quality as the output, and establishes the relationship between the input and output through

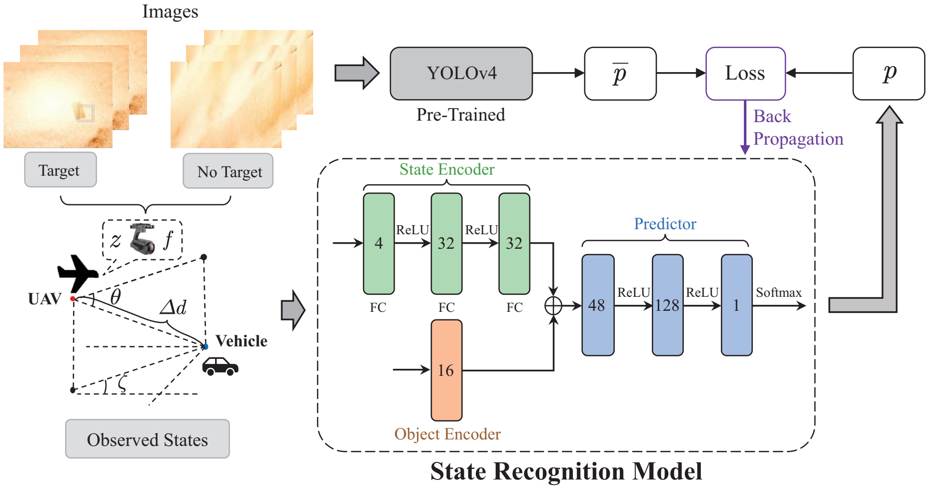

Figure 2: State recognition model (together with the observed states).

supervised learning. The observed states comprise five parameters: $[\Delta d_t, \zeta_t, \theta_t, z_t, f_t]$, where Δd_t represents the distance between the UAV and the vehicle, ζ_t illustrates the azimuth angle of the vehicle relative to the UAV, θ_t represents the pitch angle of the PTZ camera, z_t indicates the camera magnification (the camera is autofocus). In addition, f_t functions as a status flag to distinguish whether the object is present in the image, assigning a value of 1 if the object exists and 0 otherwise.

As shown in Figure 2, the network structure of the state recognition model is comprised of three parts: a state encoder, an object encoder and a predictor. The state encoder processes the observed states through three fully connected layers and returns a 32-dimensional vector. The object encoder processes the object status flag in a fully connected layer and outputs a 16-dimensional vector. The predictor joins the two vectors outputted by the state and object encoders, respectively, and feeds them into three fully connected layers to generate the object recognition probability. Moreover, except for the softmax activation function connected to the last layer of the predictor, the ReLU activation function is used between the rest of the layers. Furthermore, the number of neurons in each network layer is present in Figure 2.

We collect 24,000 images with the corresponding observed states from the UE simulator and train the state recognition model using supervised learning based on pre-trained YOLOv4. Specifically, we take the images as the input and use the object recognition probability $\bar{p}$ generated by YOLOv4 as the supervision signal, combined with the object recognition probability p generated by the state recognition model, to form the loss function[*].

To improve the stability and convergence speed of the learning process, a gradient clipping approach with a threshold of 0.5 was conducted to dynamically adjust the learning rate[†]. The neural network parameters of the state recognition model were optimized using

[*] $Loss = \frac{\sum_{i=1}^{N}|p-\bar{p}|}{N}$, where N represents the batch size of training

[†] The rule for updating the learning rate: $lr_{epoch} = \frac{1}{1+0.02\times epoch}$, where *epoch* represents the number of iterations

the Adam optimizer during the training process, and the state recognition model showed convergence after 30 iterations.

3.3. Reinforcement Learning Module

Active object tracking keeps the object within the field of view by continuously controlling the motion of the PTZ cameras. Such a process can be formulated as a classic reinforcement learning problem, and an improved proximal policy optimization (PPO) reinforcement learning algorithm is employed as an agent. The parameterization of the Markov Decision Process (i.e., state space, action space, and reward shaping) and the network architecture are described below.

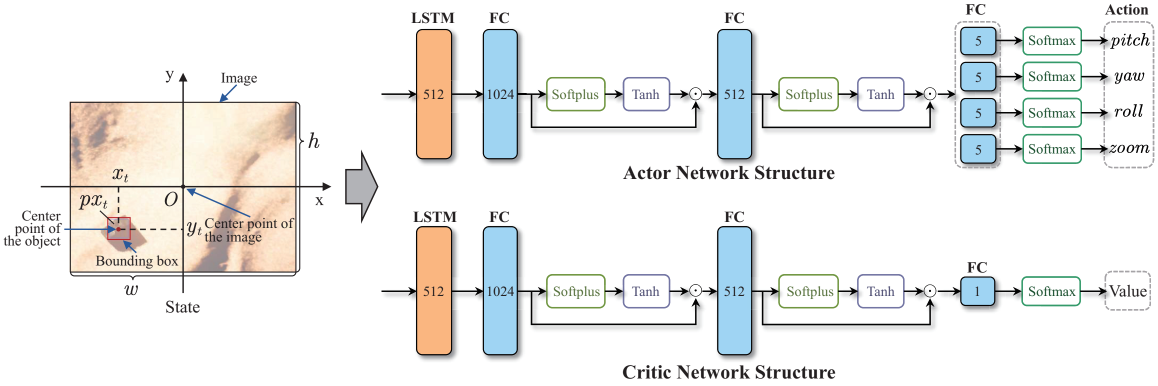

Figure 3: Reinforcement learning module.

The state space s_t at the moment t can be defined as given below:

$$s_t = \left[\frac{x_t}{w}, \frac{y_t}{h}, \frac{z_t}{z_{\max}}, \frac{px_t}{(w \cdot h)}, f_t \right]^T ,$$

where x_t and y_t donate the coordinates of the object concerning the center of the field of view, w and h represent the width and the height of the image, respectively (as shown in Figure 3). In addition, px_t represents the pixel area of the object, z_t and z_{max} represent magnification and maximum magnification. In particular, when the object is lost at the time t, s_t is set to $\left[0, 0, \frac{z_t}{z_{\max}}, 0, 0\right]^T$. In this paper, the values of w, h, z_{max} are 1024, 768 and 400x, respectively.

At the time t, we define the $action_t$ as $[pitch_t, yaw_t, roll_t, zoom_t]$, where $pitch_t, yaw_t, roll_t$, and $zoom_t$ are integers with values ranging from -2 to 2, and represent the actions of the PTZ camera's pitch angle θ, yaw angle ψ, roll angle ϕ, and camera magnification z, respectively. When executing the $action_t$, the camera state at the time $t-1$, i.e., $[\theta_{t-1}, \psi_{t-1}, \phi_{t-1}, z_{t-1}]$, is added with increments $\alpha \cdot [pitch_t/z_{t-1}, yaw_t/z_{t-1}, roll_t/z_{t-1}, zoom_t \cdot \beta]$ to obtain the required camera state at the time t, and adjust the PTZ camera. The α and β are coefficients.

In the proposed active object tracking process, the shaping of the reward function involves two aspects. First, the agent should actively perform actions to improve the object recognition probability p. Second, the object centroid should be as close to the image center as possible to achieve stable object tracking. Therefore, the final reward function consists of two parts: the knowledge-guided reward r^k and the visual distance reward r^v:

- For the knowledge-guided reward r^k, the output p of the state recognition model is introduced as prior knowledge. The actions of increasing p are performed to improve the image quality when the object is in the image; while, when the object is not in the image, reward guidance by knowledge is not feasible. Therefore, r_t^k at time step t can be obtained by:

$$r_t^k = \begin{cases} p_t, & f_t = 1 \\ 0, & f_t = 0. \end{cases}$$

- For the visual distance reward r^v, the Euclidean distance between the object centroid and the image center is introduced to motivate the agent to continuously control the PTZ camera to keep the object in the image center. The agent receives a time penalty of -1 when the object is not in the image. Thus, r_t^v at time step t can be given by:

$$r_t^v = \begin{cases} -\sqrt{\left(\frac{x_t}{w}\right)^2 + \left(\frac{y_t}{h}\right)^2}, & f_t = 1 \\ -1, & f_t = 0. \end{cases}$$

Thus, the total reward at time step t is:

$$r_t = m r_t^k + n r_t^v,$$

where m and n are the scale factors used to limit the total accumulated reward.

We propose an improved training procedure for the PPO algorithm to address the challenge of recovering the tracking process when the object has been lost for a long time. The core idea is to accumulate the rewards r_t for different moments satisfying certain criteria into a variable r_{sum} after each interaction between the agent and the environment, i.e., the world model. Then, the current episode is terminated when the value of r_{sum} falls below a predefined threshold, and a new round of training is started.

The actor and critic in the PPO algorithm are represented as neural networks, with the structures shown in Figure 3. In particular, the output layer of the actor network has four parallel fully-connected networks with five neurons each, corresponding to the four actions passed through a softmax activation function. The critic network has a similar structure to the actor network, except that it has only one neuron in the output layer that directly outputs the Q value. A Block composed of softplus and tanh activation functions is used after the second and third layers, which makes the networks easier to optimize.

3.4. World Model

Reinforcement learning requires iterative optimization through continuous interaction with mass data in the environment. For this purpose, we conduct a UE simulator to emulate a realistic desert environment with mild undulating terrain, forests, vehicles and UAVs[‡]. As a simulated environment built on the UE engine, the UE simulator provides realistic

[‡]. The movement of UAVs and vehicles follows the laws of physics.

images, high-fidelity simulated parameters, and flexible experimental settings for reinforcement learning training. However, the high demand for computational resources and the inefficient communication mechanism of the UE simulator hinder the efficiency of model training. Therefore, we construct a world model (Ha and Schmidhuber, 2018) based on environment abstraction to replace the UE engine, which can provide crucial parameters involved in the simulation environment and significantly improve training efficiency.

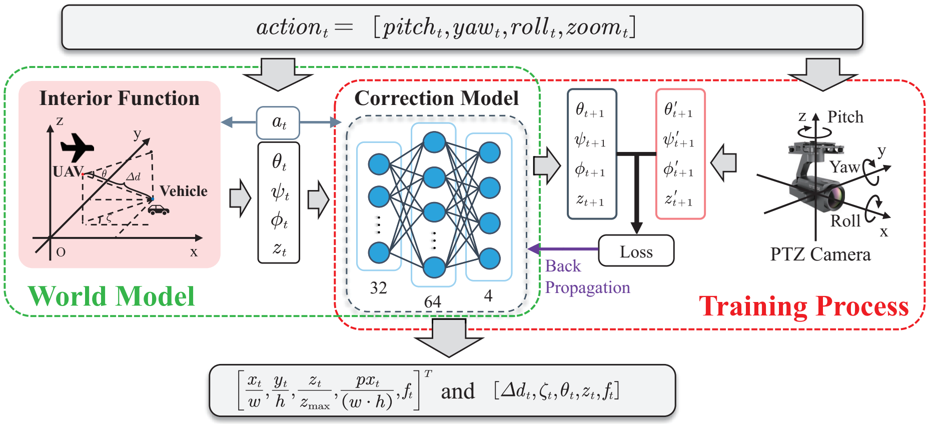

Figure 4: World model.

As shown in Figure 4, the world model is composed of two components, i.e., the interior function and the correction model. In the UE simulator, we can directly obtain parameters such as the positions of the vehicle and the UAV, and the attitudes of the PTZ camera. While using the world model instead of the UE simulator, we need to use the interior function to store and compute these essential parameters during training.

According to the complexity of the calculation when obtaining parameters from the world model, we divide the parameters into two categories: (1) parameters that can be inferred from the speed and directions of the vehicle and UAV, such as the positions of the vehicle and UAV, and the attitude of the PTZ camera; (2)the acquisition of the parameters requires the projection of the vehicle's coordinates in the world coordinate system onto the pixel coordinate system obtained by the PTZ camera, such as the coordinates of the object from the center of the field of view, and the pixel area of the object (i.e., x_t, y_t and px_t as mentioned in Section 3.3).

The method of calculating the latter parameters is described below. Briefly, the vehicle position is first converted from the world coordinate system to the PTZ camera's coordinate system based on the position relations of the vehicle, the UAV and the PTZ camera in the world coordinate system, projected onto the image plane of the PTZ camera to obtain the normalized coordinates. Then, the vehicle position is converted to the location coordinates (x_t and y_t) in the pixel coordinate system based on the camera's intrinsic matrix K. Finally, the pixel area of the object (px_t) is obtained from the conversion of the vehicle position in the pixel coordinate system. The detailed calculation procedure is as follows.

- Since the PTZ camera coordinates stored in the interior function are in the UAV coordinate system, the rotation matrix R_c^u of the camera coordinate system to the

UAV coordinate system and the rotation matrix R_u^w of the UAV coordinate system to the world coordinate system need to be calculated.

$$R_c^u = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_u^w = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

- According to the positions of the vehicle P_v, UAV P_u, and the PTZ camera relative to the UAV $P_{cam2uav}$, the vehicle position P_v is converted to the position in the camera coordinate system, i.e., P_c.

$$P_c = (R_c^u)^{-1} \left((R_u^w)^{-1} (P_v - P_u) - P_{cam2uav} \right)$$

- Normalize P_c and multiply it by the PTZ camera's intrinsic matrix K to obtain the coordinates P_p of the vehicle in the pixel coordinate system.

$$P_p = K \cdot \frac{P_c}{\max(P_c[2], 1e-5)},$$

  where $P_c[2]$ donates the third parameter of P_c. Note that we have set a precaution in the formula to prevent the denominator from being zero.

- The pixel area of the object px_t can be composed of a region consisting of four points selected at a distance of 2 meters each in front, back, left and right of the vehicle position P_v (determined in the above three steps).

The correction model fits the control error between the interior function and the UE simulator, which enables the world model to achieve a similar control performance as the UE simulator. The correction model adopts a three-layer fully connected network structure (as shown in Figure 4), taking the PTZ camera's state (i.e., angle and magnification) and the corresponding actions provided by the agent as inputs and the next state of the PTZ camera as the output. We extracted 8,000 datasets (including the input and the output shown in Figure 4) from the UE simulator, and trained the correction model using supervised training with a learning rate of 0.2. To optimize the neural network parameters of the correction model, we use the Adam optimizer with the mean absolute error as the loss function:

$$Loss = \frac{\sum_{i=0}^{N} \left(\left| \theta_i - \theta_i^{'} \right| + \left| \psi_i - \psi_i^{'} \right| + \left| \phi_i - \phi_i^{'} \right| + \left| z_i - z_i^{'} \right| \right)}{N}.$$

Moreover, the correction model shows a convergence trend after 30 iterations.

Finally, we initialize the actor and critic parameters, interact with the world model to obtain the training data and store it in a replay buffer in the training process. Then, we sample a mini-batch of 256 from the replay buffer as our dataset and apply Adam (the learning rate of actor and critic is $1e-4$ and $2e-4$) to optimize the network. The generalized advantage estimation (GAE) parameter λ and the clipping ϵ are set to 0.95 and 0.2, respectively. The maximum number of global episodes is 8K and the maximum number of steps N is 400. For the reward function, the values of the γ, m and n are 0.99, 0.1 and 0.1, and the action coefficient α, β are 50 and 5, respectively. We use TensorFlow as a deep learning framework to train the actor and critic networks with a PC containing an AMD Ryzen 7-5800H (3.20 GHz ×16) processor, 16 GB of RAM, and an NVIDIA RTX 3050 with 4 GB of VRAM.

4. Experimental Setup and Results Analysis

4.1. Baseline and Evaluation Criteria

As the baseline approaches, we compare our method with the following three state-of-the-art methods proposed by other researchers in similar tasks.

- Active Object Tracking (AOT) Xi et al. (2021) method can effectively control UAVs to track moving objects in realistic scenarios.
- C^{3}Net approach Kyrkou (2021) introduces a deep convolutional camera controller neural network, which converts visual information into camera motion and enables real-time monitoring of camera target tracking.
- PID controller is a well-established and widely adopted control method employed in diverse domains. Considering the effect of the PTZ camera's control response characteristics and the vibration noise encountered during tracking, the PID coefficients are manually adjusted for the tracking task,.

Regarding the aforementioned three baseline methods, the output control actions include incremental adjustments of the PTZ camera's three-axis (pitch, yaw, and roll) angles during model training and testing. Furthermore, the magnification of these methods was set to 50 times.

The performance evaluation includes the following four criteria:

- **Stability.** The stability of the tracking process is measured in terms of center location error, which represents the Euclidean distance (in pixels) between the object centroid and the image center in a step. Continuous smaller values of center location errors indicate better stability.
- **Robustness.** Ro is used to evaluating the robustness of the active tracker, which is the percentage of frames in which the tracker loses the object during the tracking process. Smaller Ro means better robustness.
- **Image quality.** Object recognition probability is adopted to measure image quality, which reflects the effect of introducing the state recognition model as a knowledge-guided model to improve observation quality. Higher probability indicates better image quality obtained during the object tracking process.

- **Training speed.** During the model training, the time for the reward values to stabilize reflects the training speed, and also reflects the effect of introducing the world model as a knowledge-guided model. Shorter time means faster training.

4.2. Experiments and Results

To conduct the experiments, we randomly initialized the starting position and orientation of the vehicle and the UAV, with the vehicle moving at 12m/s and the UAV flying at 300m altitude. Initially, the camera was set at a magnification of 50x and precisely aimed at the vehicle.

Stability. We compared the object tracking stability for three vehicle motions, i.e., rectilinear, S-curve and random. To further verify the tracking stability, we added three disturbance modes to the vehicle and the UAV. Regarding the vehicle disturbance, the direction turns arbitrarily and the speed changes randomly in the range of 0 to 20m/s. For the UAV disturbance, we applied a slight vibration to the PTZ camera by setting random changes in the pitch and roll angles, which caused the object to vibrate within the camera viewfinder frame. Moreover, we compared the results obtained in each scenario with the tracking performance of the other three methods.

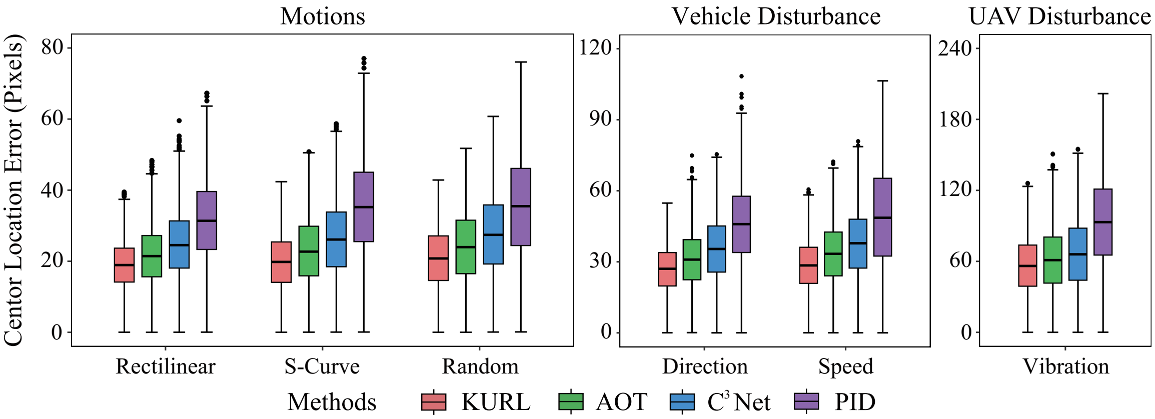

Figure 5: Comparison of the tracking stability obtained in different scenarios.

Figure 5 shows a series of box plots illustrating the distribution of center location errors for four methods, i.e., our proposed KURL model and the other three baseline methods. The evaluation covers 30 episodes (12,000 steps) for each scenario mentioned above. According to Figure 5, the center location errors of the proposed KURL model are consistently the smallest across all scenarios, followed by the AOT and C^{3}Net methods, while the PID method performs the worst. To further evaluate the significance of the difference for each scenario, we calculated the Wilcoxon Signed Rank test Rosner et al. (2006) on the center location errors of the four methods. The results of the statistical tests indicate a significant difference (i.e., p-value <0.05) in the center location errors between the KURL model and the other three methods in each scenario. Thus, **the KURL model is significantly more stable than the other three baseline methods in object tracking across all proposed scenarios, which encompass various vehicle motions and disturbance modes.**

Table 1: Ro values in different scenarios.

Scenarios	KURL	AOT	C^{3}Net	PID
inFoV	**0.12%** (6)	0.15% (9)	0.26% (8)	0.20% (9)
outFoV	**2.84%** (44)	11.2% (41)	48.49% (42)	95.69% (41)
Total	**2.51%**	**9.25%**	**40.77%**	**78.50%**

Robustness. We evaluated the robustness of the proposed KURL model by comparing the tracking performance after losing the object (due to various reasons, e.g., interference or occlusion) during the UAV flight with the other three baseline methods. We set the object vehicle to move randomly within an episode (400 steps) and to be lost (i.e., no longer receiving the tracking signals) at step 100. After the loss, the movements of the vehicle and the UAV remain constant, and an attempt is made to re-observe and re-track the object at step 120. At that moment (step 120), there are two scenarios can be observed: 1) the object is still in the field of view ($inFoV$) and 2) the object disappears in the camera viewfinder frame, i.e., out of the field of view ($outFoV$).

Table 1 shows the number of occurrences (numbers in parentheses) for $inFoV$ and $outFoV$ scenarios after 50 experiments using the four methods, respectively, and the corresponding average values of Ro, i.e., the percentage of frames where the object was lost after step 120. In addition, the "Total" row represents the average Ro values in a total of 50 experiments. From Table 1, $outFoV$ has a much higher probability of occurrence than $inFoV$, which means **the object has a lower chance of reappearing in the field of view after being lost even for a short period of time** (e.g., 20 steps).

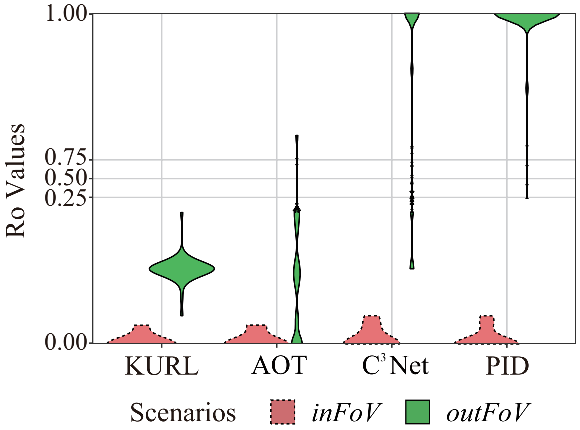

Figure 6: Comparison of the tracking stability obtained in different scenarios.

Moreover, Table 1 also reveals that much lower Ro values can be obtained using our method, especially in the $outFoV$ scenario, which means that the KURL model is more robust. It can be further visualized in Figure 6, which compares the change in Ro values for each method in the $inFoV$ and $outFoV$ scenarios by the violin plots. The more elongated the shape of the violin, the larger the variance in the corresponding group; and the wider the violin plot, the higher the density. By observing Figure 6, we note that the Ro values of the KURL model in the *inFoV* scenario are slightly lower than those of the baseline methods. In stark contrast, the Ro values of the KURL model in the *outFoV* scenario are surprisingly smaller with a more concentrated distribution. The results highlight that **the KURL model is significantly better than the other methods in terms of robustness, especially when re-tracking after object loss.**

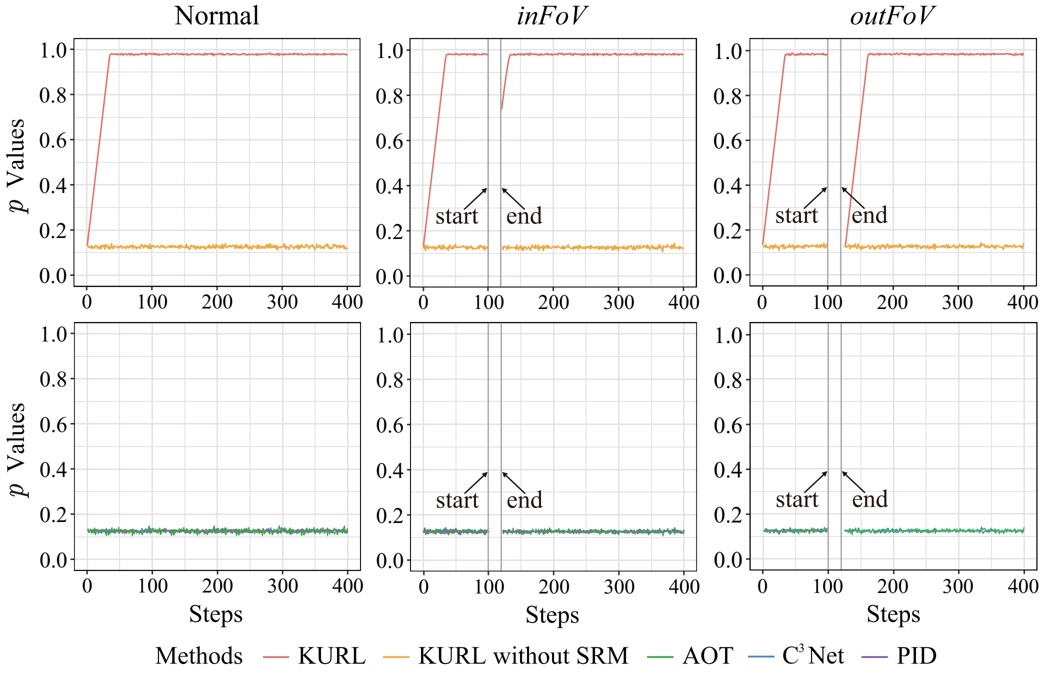

Figure 7: Comparison of the image quality, i.e., the object recognition probability (p values), in different scenarios. The “start” and “end” represent the time step at which the object starts to be lost and the observation is resumed, respectively.

Image Quality. At the initial magnification (50x) of the PTZ camera, we conducted experiments with the KURL model, the KURL model without the state recognition model (abbreviated as “KURL without SRM” in the figure), and the other three methods in different scenarios, and the results are shown in Figure 7. In the normal object-not-loss scenario (“Normal”) and the first 100 steps of the object-loss scenarios (“*inFoV*” and “*outFoV*”), the KURL model without the state recognition model and the other three baseline methods all maintain an object recognition probability of approximately 0.124 following calculation. However, our proposed KURL model increases the object recognition probability to nearly 1 in a short period of about 40 steps by controlling the magnification (zoom in to approximately 400x) of the PTZ camera.

After step 120, the KURL model shows a clear advantage in improving the quality of the observed images in the object-loss scenarios (i.e., *inFoV* and *outFoV* subfigures in Figure 7). For the *inFoV* scenario, all five methods can consistently observe the object recognition probability since the object is still in the field of view. Among them, the KURL model can quickly improve the image quality by continuously controlling the magnification of the PTZ camera, while the other four methods can only maintain around the initial p value.

Regarding the $outFoV$ scenario where the object disappears in the camera viewfinder frame, the KURL method and the other three methods (except the PID controller) can achieve object relocation based on historical information, and thereby re-judging the image quality. In addition, the KURL model can quickly recover a high quality of the observed images again by conducting the state recognition model to control the PTZ camera. However, KURL without SRM, AOT and C^{3}Net methods can only maintain the object recognition probability around the initial p value. The results indicate that **the state recognition model can effectively guide the reinforcement learning method to improve the observed image quality during active object tracking.**

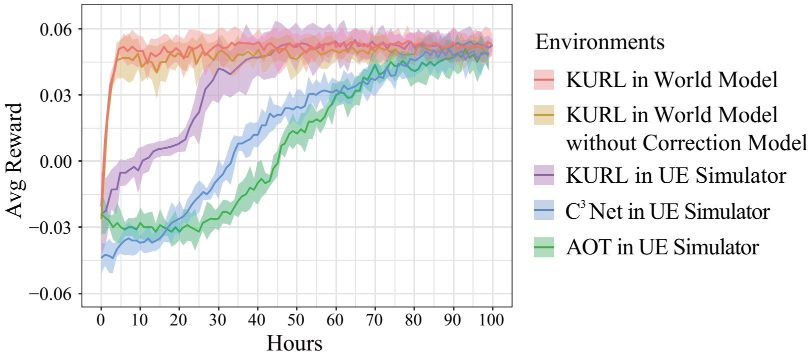

Figure 8: Comparison of the training speed in different environments and the shaded area indicating one standard deviation within each time bin.

Training Speed. To assess the impact of the world model on training speed, we conducted the training process of the various methods in different environments (using the world model or the UE simulator), as depicted in Figure 8. Initially, the KURL model can reach stability after about 45 hours of training in the UE simulator. In contrast, the C^{3}Net and AOT methods require at least 80 and 100 hours, respectively, highlighting that our method is roughly twice as fast as these two methods in achieving stability. While in the world model, the KURL model attains the same stable reward in a mere 4.5 hours, which is approximately 10 times faster than training in the UE simulator. Furthermore, we compared the impact of having a correction model on the performance of the KURL model. Figure 8 shows that the KURL model (with the correction model incorporated) can obtain a superior average reward with almost no impact on the training speed. The above results suggest that **our proposed method can be trained more rapidly in the UE simulator compared to the baseline methods. In addition, replacing the UE simulator with the world model can greatly enhance the training speed of the KURL model, while adding the correction model to the world model further improves the tracking performance.**

5. Conclusion

This paper proposes a novel Knowledge-gUided Reinforcement Learning (KURL) model for the high-altitude tracking environment. The method consists of two embedded knowledge-guided models (i.e., the state recognition model and the world model), together with a reinforcement learning module. Firstly, we designed a state recognition model that acts as a knowledge module in the reward function to guide the reinforcement learning algorithm to actively control the PTZ camera, which can improve the observed image quality. Secondly, a memory-enabled actor-critic network structure was designed to address the problems of interference and target loss. Lastly, a world model was used to substitute the UE simulator as to improve the training efficiency. The experimental results indicate that the KURL model improves image quality while also enhances the robustness and stability compared with the other state-of-the-art active object tracking methods.

However, due to the limitations of the experimental conditions, future work will focus on deploying the KURL model in real environments, considering designing more knowledge modules to address other interfering factors of object tracking, and further enhancing the applicability of the KURL model.

References

Yanyu Cui, Biao Hou, Qian Wu, Bo Ren, Shuang Wang, and Licheng Jiao. Remote sensing object tracking with deep reinforcement learning under occlusion. *IEEE transactions on geoscience and remote sensing*, 60:1–13, 2021.

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

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Christos Kyrkou. C^3net: end-to-end deep learning for efficient real-time visual active camera control. *Journal of Real-Time Image Processing*, 18:1421–1433, 2021.

Bo Li, Zhi-peng Yang, Da-qing Chen, Shi-yang Liang, and Hao Ma. Maneuvering target tracking of uav based on mn-ddpg and transfer learning. *Defence Technology*, 17(2): 457–466, 2021.

Jing Li, Jing Xu, Fangwei Zhong, Xiangyu Kong, Yu Qiao, and Yizhou Wang. Pose-assisted multi-camera collaboration for active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 759–766, 2020.

Siyi Li, Tianbo Liu, Chi Zhang, Dit-Yan Yeung, and Shaojie Shen. Learning unmanned aerial vehicle control for autonomous target following. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4936–4942, 2018.

Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. In *International conference on machine learning*, pages 3286–3295. PMLR, 2018.

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

Yongle Luo, Kun Dong, Lili Zhao, Zhiyong Sun, Erkang Cheng, Honglin Kan, Chao Zhou, and Bo Song. Calibration-free monocular vision-based robot manipulations with occlusion awareness. *IEEE Access*, 9:85265–85276, 2021.

Xiaojuan Ma, Yanhui Wang, Shaoqiong Yang, Wendong Niu, and Wei Ma. Trajectory tracking of an underwater glider in current based on deep reinforcement learning. In *OCEANS 2021: San Diego–Porto*, pages 1–7. IEEE, 2021.

Payal Mittal, Raman Singh, and Akashdeep Sharma. Deep learning-based object detection in low-altitude uav datasets: A survey. *Image and Vision computing*, 104:104046, 2020.

Jiseon Moon, Savvas Papaioannou, Christos Laoudias, Panayiotis Kolios, and Sunwoo Kim. Deep reinforcement learning multi-uav trajectory control for target tracking. *IEEE Internet of Things Journal*, 8(20):15441–15455, 2021.

Bernard Rosner, Robert J Glynn, and Mei-Ling T Lee. The wilcoxon signed rank test for paired comparisons of clustered data. *Biometrics*, 62(1):185–192, 2006.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Pavlos Tiritiris, Nikolaos Passalis, and Anastasios Tefas. Temporal difference rewards for end-to-end vision-based active robot tracking using deep reinforcement learning. In *2021 International Conference on Emerging Techniques in Computational Intelligence (ICETCI)*, pages 21–25. IEEE, 2021.

Yu Wang, Chong Tang, Shuo Wang, Long Cheng, Rui Wang, Min Tan, and Zengguang Hou. Target tracking control of a biomimetic underwater vehicle through deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3741–3752, 2021.

Mao Xi, Yun Zhou, Zheng Chen, Wengang Zhou, and Houqiang Li. Anti-distractor active object tracking in 3d environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3697–3707, 2021.

Haohui Zhang, Pingkuan He, Ming Zhang, Daqing Chen, Evgeny Neretin, and Bo Li. Uav target tracking method based on deep reinforcement learning. In *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pages 274–277. IEEE, 2022.

Wenlong Zhao, Zhijun Meng, Kaipeng Wang, Jiahui Zhang, and Shaoze Lu. Hierarchical active tracking control for uavs via deep reinforcement learning. *Applied Sciences*, 11(22): 10595, 2021.

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