Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20134-w

Meta-reinforcement learning for active visual tracking about space non-cooperative object

Zhongliang Yu[1]

Received: 11 February 2024 / Revised: 26 June 2024 / Accepted: 18 August 2024

Abstract
This paper addresses the challenge of active tracking of space non-cooperative targets, a critical task in various aerospace applications. Traditional active tracking algorithms often require extensive data and suffer from limited generalization ability, making them inefficient for tracking targets with diverse characteristics. To overcome these limitations, we propose an end-to-end active target tracking method named Meta-Reinforcement Learning based Active Visual Tracking (MRLAVT). This approach integrates meta-reinforcement learning, enabling the system to quickly adapt to new tasks by leveraging experiences from previous tasks. By employing convolutional neural networks to extract information from images and generate corresponding actions, MRLAVT demonstrates strong adaptability and robustness in tracking targets with varying characteristics. Experimental results confirm the effectiveness of our proposed algorithm, showcasing superior performance in scenarios involving both few adaptations and non-adaptation. Overall, MRLAVT significantly reduces the complexity of system integration while achieving high-quality tracking results.

Keywords Active object tracking · Meta-reinforcement learning · Space non-cooperative

1 Introduction

In recent years, advancements in computational processing capabilities have led to significant progress in visual tracking algorithms, greatly improving processing speed and tracking accuracy. However, most current visual tracking methods rely on passive tracking approaches, where the position of the chaser remains fixed. This assumption simplifies the tracking problem by assuming that the target remains within the field of view, thereby reducing the complexity of the tracking task.

In practical applications, however, targets are often in motion and may move out of the field of view. For instance, when tracking a target using a mobile robot or drone, it is crucial to continuously adjust the chaser's perspective to ensure that the target remains within the

✉ Zhongliang Yu
zlyu@cqu.edu.cn

[1] School of Automation, Chongqing University, Shazhengjie 174, Chongqing 400044, Chongqing, China

center of view. Failing to do so could result in the target moving out of sight if the chaser's position remains fixed.

To address these challenges, we propose a meta-reinforcement learning method for solving the active object tracking problem, as illustrated in Fig. 1 In this approach, the active tracker generates corresponding actions directly from the observed image. Leveraging the generalization capability of meta-learning, our method enables more effective active tracking, allowing for quick adaptation to new tracking targets while maintaining high performance. Importantly, when tracking a new target, the MRLAVT algorithm does not require retraining, thereby saving considerable training time.

We simulate the space environment in CoppeliaSim to create an active target tracking environment, thereby saving substantial expenses associated with labeling work, training time, and trial and error costs. Within this virtual environment, the chaser observes the target, acquires its status, and then takes corresponding actions to transition to a new state. Compared to DRLAVT, our MRLAVT method demonstrates enhanced tracking performance, particularly when the target's trajectory and background change, or when the tracking target becomes blurred.

Although traditional deep learning methods are powerful, they often require extensive training from scratch, consuming significant time and computational resources to achieve satisfactory performance. Moreover, they may suffer from poor generalization when applied to new tasks or datasets. By integrating meta-learning into the active target tracking problem, we can mitigate the challenges posed by small datasets. Meta-learning enables the trained model to exhibit strong generalization capabilities, allowing it to adapt rapidly to new tasks through efficient adaptation. This not only reduces training time but also enhances the performance and efficiency of the active target tracker.

Active tracking of space non-cooperative targets holds significant importance as it forms the foundation of various space missions, including space junk capture, space docking, space

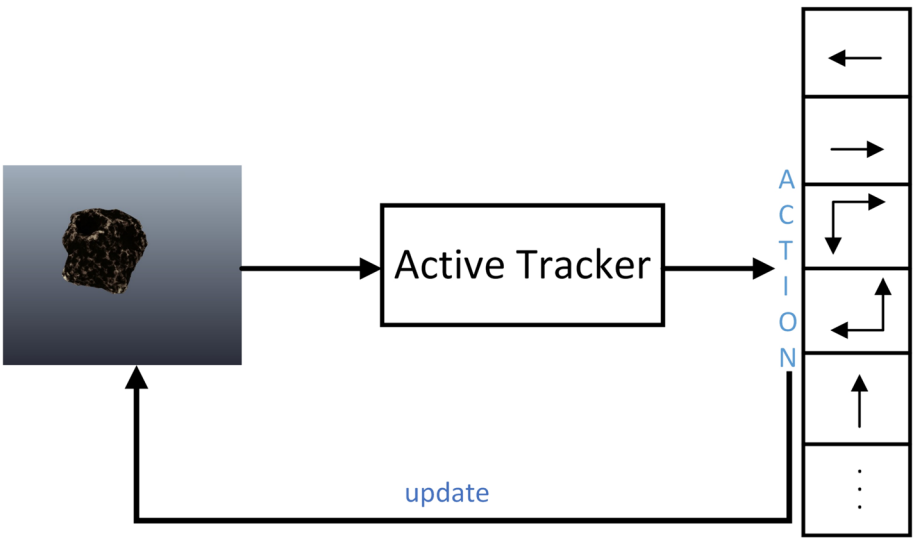

Fig. 1 The pipeline of active object tracking with space non-cooperative

exploration, and target trajectory prediction. The critical role of active tracking has garnered increased attention in recent years, making it a focal point of research efforts. Drawing from insights and algorithms developed for passive tracking, active tracking algorithms have witnessed substantial advancements, further underscoring its significance in space missions and beyond.

The paper is structured as follows: Section 2 provides an overview of related work on active tracking algorithms for space non-cooperative targets. In Section 3, we present the details of our proposed algorithm, MRLAVT. Section 4 describes the experiments conducted to demonstrate the effectiveness and advancements of MRLAVT through extensive testing. Finally, Section 5 offers concluding remarks summarizing the key findings and contributions of the paper.

2 Related works

Given the focus of our paper on active object tracking, meta-reinforcement learning, and space non-cooperative objects, it is imperative to provide a comprehensive review of existing literature on these topics. We will present a thorough examination of previous work, highlighting key contributions and insights in each area.

2.1 Active object tracking

Object tracking encompasses both active and passive tracking methods [1–3]. In recent years, passive tracking techniques have witnessed significant advancements in terms of both accuracy and speed. Various approaches have been proposed to address challenges such as occlusion, blurring, changes in lighting conditions, and target deformation, which commonly occur during the tracking process [4–8]. Early tracking algorithms predominantly relied on optical flow methods [9], filtering techniques [10], and kernel-based algorithms [11]. However, these methods often encountered challenges in implementation and struggled to maintain stable accuracy. In contrast, contemporary tracking algorithms can be broadly categorized into filtering-based [12] approaches and deep learning-based methods [13].

In recent years, active object tracking has seen significant advancements. Li [14] extended active object tracking from single-camera setups to multi-camera scenarios. In their approach, multiple cameras track the target cooperatively by sharing camera poses, thereby addressing challenges associated with active target tracking. Wang [15] combined prediction information from the Kalman filter with information recognized by neural networks to calculate target orientation and distance. Luo [16, 17] proposed an end-to-end solution based on deep reinforcement learning for active object tracking, utilizing ConvNet-LSTM architectures to generate corresponding actions. Xi [18] introduced an innovative end-to-end anti-distractor active target tracking framework. Tian [19] developed and implemented a binocular vision system for actively tracking moving targets. Zhong [20] proposed a hybrid cooperative-competitive multi-agent approach to confront trackers, enhancing the robustness of active tracking systems. Noel [21] proposed an active system based on mirrors, enabling changes in the camera's viewing angle to achieve active tracking of underwater targets. Zhou [22–24] focused on realizing active tracking of space targets in virtual simulation environments.

2.2 Meta-reinforcement learning

Meta-learning, also known as learning to learn, goes beyond simply learning to perform specific tasks; it involves learning the underlying principles of problem-solving. Meta-learning enables rapid adaptation to new tasks by leveraging learned problem-solving abilities. For example, in image classification tasks [25], meta-learning not only learns to classify cats and dogs but also learns the general principles of classification problem-solving. Meta-learning is a versatile learning strategy that can be applied to various models, including convolutional neural networks, deep reinforcement learning, and more. Active visual tracking demands continuous interaction between the tracker and the environment to maintain the target in view. Reinforcement learning excels in sequential decision-making problems, making it suitable for active target tracking. To address the challenge of poor generalization ability in reinforcement learning algorithms, we employ meta-reinforcement learning to enhance the tracker's generalization capabilities and achieve superior performance in active target tracking.

In recent years, reinforcement learning [26, 27] has gained widespread adoption, thanks to advancements in computational power. However, reinforcement learning typically involves extensive trial and error, resulting in lengthy training times. The increase in computing power has significantly reduced training times, making the practical application of reinforcement learning feasible, and expanding its scope of application. Moreover, with the enhanced image processing capabilities of convolutional neural networks (CNNs) [28], CNN-based reinforcement learning has notably improved visual tracking performance [29].

The optimization-based meta-learning method [30] is a classic approach that employs a two-layer structure consisting of an inner and outer loop. Initially, the parameters of the outer layer model are duplicated to the parameters of the inner layer model. Subsequently, the inner layer network model iteratively optimizes its parameters across various tasks. After multiple optimization iterations [31], the loss value or gradient of the task on the inner model is utilized to update the external model parameters. This process is then repeated with the parameters of the outer model copied to the inner model. Notably, the loss value or gradient of the task on the inner model encapsulates rich information, encompassing not only details on how to complete the current task but also the ability to learn how to solve tasks efficiently.

The metric-based method [32], also known as the nonparametric method, primarily focuses on discerning the dissimilarities between tasks and categorizes them based on task similarity. This approach resembles clustering tasks, where tasks exhibiting similar characteristics are grouped together. Initially, data (such as images/text, etc.) in the support set is encoded, and vector representations of the data are learned. Subsequently, the data within the same category is aggregated to derive a class vector. The similarity (or distance measure) between the query set vector and the class vector is then computed, with the class exhibiting the highest similarity being selected. Siamese networks [33] and recurrence with attention mechanisms [34] are classical metric-based methods, although they may pose challenges when applied to reinforcement learning environments.

The model-based method [35–40] is a classical meta-learning algorithm. In this approach, tasks are sequentially inputted into the model, causing changes in the model's internal state. The internal states of the model can capture task-specific information, which can then be utilized to make predictions about new inputs. These predictions are based on internal dynamics that are hidden from the external environment, rendering model-based techniques as "black boxes." Given that information from previous tasks must be retained, model-based tech-

niques incorporate an in-memory component that is neither entirely internal nor external. One notable aspect of model-based methods is their flexibility, as human designers can freely choose the internal dynamics of the algorithm. Consequently, model-based techniques are not constrained to learning only good feature representations; they can also learn internal dynamics for task processing and prediction. In contrast to optimization-based methods, the optimization process for model-based techniques is simpler and does not require second-order gradients. However, it has been observed that model-based methods generally exhibit lower ability to generalize to out-of-distribution tasks compared to optimization-based methods.

We employ an optimization-based meta-learning algorithm for the active target tracking task, which offers adaptability across a wide range of tasks compared to other types of meta-learning algorithms.

3 MRLAVT algorithm

In this paper, we employ meta-learning based on deep reinforcement learning for active target tracking. Leveraging meta-learning, our tracker exhibits adaptability by quickly adapting to new tasks based on previous experiences. With the initial model as a foundation, minimal adaptation steps yield excellent tracking results across a variety of targets.

3.1 Problem formulation and task details

The objective of active visual tracking is to navigate the environment, detect the target, and take appropriate actions based on input frames to keep the target within the field of view. This problem can be formulated as a classic reinforcement learning (RL) task. We define the tracker as an RL agent that iteratively interacts with the environment, receiving observation inputs, generating rewards, and selecting actions. In our scenario, the chaser serves as our agent, observing the state of the tracking target. The captured images represent a first-person perspective, providing only partial information about the true hidden state.

We assume that both the tracker and the target have freedom of movement in 3D space. The tracker receives real-time image frames from its front-facing camera, and its action space consists of discrete movements: *don't move , forward , right , left , backward , forward&right , forward&left , backward&right , backward&left , up , down*. At the start of each episode, both the tracker and the target are randomly positioned within the environment. While the initial position of the target impacts the tracking outcome, our goal is to maintain the target within the tracker's field of view, with the reward function designed to prioritize the target's visibility for maximum reward. Understanding the visual appearance characteristics of the target is crucial for effective tracking.

3.2 Virtual environment specification

We leveraged CoppeliaSim's robust virtual environment simulation capabilities to construct 18 models of space non-cooperative targets, categorized into five types: (1) Asteroids, (2)

Return capsules, (3) Rockets, (4) Satellites, and (5) Space Stations (see Fig. 2). These targets vary in physical dimensions, shape, and other attributes, ensuring the diversity and accuracy of our dataset. The inclusion of a wide range of targets is crucial for validating the robustness and generalization capabilities of our algorithm (Fig. 3).

Due to space constraints, we refrain from detailing the conversion between different coordinate systems and specific settings within the virtual environment. The initial position of the target within the tracker's field of view significantly impacts tracking accuracy. We assume the initial state of the target is observable, and it appears in our field of view in a regular state as the initial condition. To enhance algorithm robustness, we distribute the initial position of the target within the field of view according to a normal distribution.

3.3 Meta-reinforcement learning for active visual tracking

We employed the classical reinforcement learning algorithm Deep Q-learning (DQL) as the foundational framework for interacting with the environment. DQL utilizes conventional convolutional neural networks (CNNs) to extract target-related information from color or depth images, facilitating optimal action decisions. The incorporation of meta-reinforcement learning further enhances the generalization capability of reinforcement learning in active target tracking. The internal structure of our Meta-Reinforcement Learning for Active Visual Tracking (MRLAVT) algorithm is illustrated in Fig. 4

Our action library comprises 11 actions, each of which triggers a transition to a new state within the environment and yields a corresponding reward value. We utilize deep neural networks to approximate the optimal action function. To evaluate the algorithm's performance across various network architectures, we employed ResNet, a variant CNN, as our experimental network structure [41]. The experimental results demonstrate the algorithm's robust performance even under complex network configurations.

As the iteration progresses, repeatedly selecting actions randomly fails to leverage the learning acquired over time. One approach to address this issue is to consistently choose the action with the highest current Q-value after the first iteration. However, this strategy risks falling into a local optimum, as there may exist better actions yet to be explored. To balance exploration and exploitation, we employ an ε-greedy policy. Under this policy, there is a probability of ε to explore by randomly selecting an action, and a probability of $(1 - \varepsilon)$ to exploit by choosing the action with the highest current Q-value.

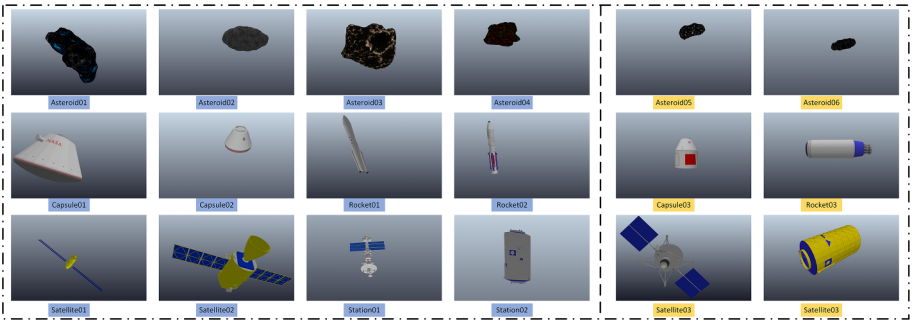

Fig. 2 The 5 types space non-cooperative target models. The model on the left is used for training and the right is used for testing

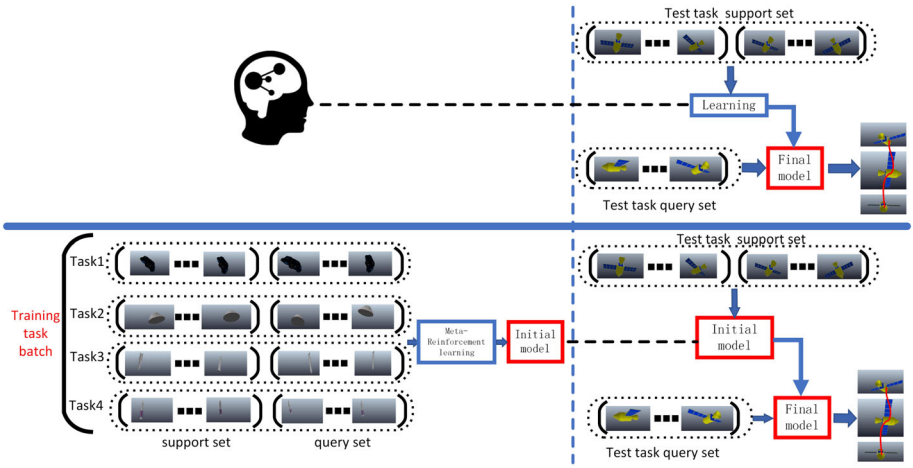

Fig. 3 The pipeline of meta-learning about visual tracking. Top: the process of humans to complete visual tracking. Bottom: contrarily, meta-learning learn multiple visual tracking task then become a good learner, and then complete the active visual tracking

Meta-reinforcement learning acquires the capacity to learn by processing numerous tasks, enabling it to excel not only in mastering training tasks but also in adapting effectively to new tasks. The initial meta-reinforcement learning model demonstrates strong generalization capabilities, eliminating the need for training from scratch and thereby significantly reducing training time. Figure 3 illustrates the application of meta-reinforcement learning to active object tracking.

Meta-learning operates on a task-based paradigm, where each task consists of two key parameters: n-way and k-shot. Here, n-way denotes the number of distinct types of tracking targets, while k-shot signifies the selection of k training instances for each target type from the available tracking target data. Throughout the training process, a diverse array of new training data is continuously generated.

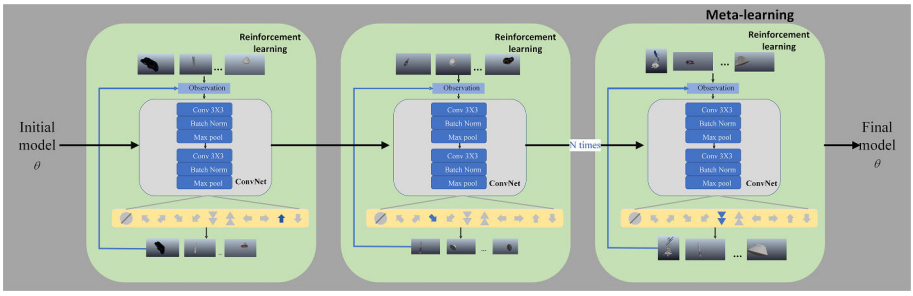

Fig. 4 The internal structure of MRLAVT about active visual tracking

In our paper, we set several key parameters: the maximum episode length (L) is set to 1000, the total number of episodes (EN) is 300, and the initial buffer size is 10,000. We utilize a simulated virtual environment provided by CoppeliaSim. Our model, represented by a parametrized function f_θ with parameters θ, adapts to a new task $\mathcal{T}i$ by updating its parameters to θi. This update is achieved through one or more gradient descent iterations on task $\mathcal{T}_i$. For instance, we employ a single gradient update in our method.

Algorithm 1 Meta-reinforcement learning for active visual tracking.

Input: Initial Active Tracking Target, α, β: step size hyperparameters
  Meta-parameters θ, Initial Target State s1;
  for i< initial buffer;
    Action=$\theta(s1)$;
    s2, Reward=$env(s1)$;
    Store(s2, Reward, s1, Action) in Memory;
while episode number <EN;
    Sample batch of train tasks $\mathcal{T}_i$ from Memory;
    for all $\mathcal{T}_i$ **do**;
      Sample datapoints $\mathcal{D} = (state, action)$in $\mathcal{T}_i$;
      Compute adapted parameters with gradient
      descent: $\theta_i' = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$;
    end for
    Sample datapoints $\mathcal{D} = (state, action)$;
    Update $\theta \leftarrow \theta - \beta\nabla_\theta \sum_{\mathcal{T}_j} \mathcal{L}_{\mathcal{T}_j}\left(f_{\theta_i'}\right)$
    if done=**True**
      break;
  end while
end while

When updating the parameters of the inner layer model, we utilize the function f_θ to represent the model with parameters θ. Throughout the training process, the model's parameters are transformed to θ_i. In our approach, the parameter vector θ_i undergoes one or more gradient descent updates specific to the task. For instance, when employing a gradient update:

$$\theta_i' = \theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \tag{1}$$

Where α represents the learning rate for updating the inner layer parameters, and conducting multiple updates to these parameters can enhance the tracker's performance.

Our tasks consist of n-way k-shot samples from datasets used to optimize the parameters of the inner network model. More specifically, the meta-goals are as follows:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right) \tag{2}$$

After completing the update of the inner layer parameters, we proceed to update the parameters of the outer layer model. The update of the outer layer model parameters should utilize either the loss value or the gradient of the inner model parameters on the query set.

The specific update procedures are as follows:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i} \left(f_{\theta_i'} \right) \tag{3}$$

Where β represents the meta-learning rate. In essence, the complete algorithm can be seen in Algorithm 2. It's important to note that the meta-optimization is conducted on the model parameter θ, while the loss value or gradient is obtained by computing the model parameter θ_i of the inner layer. The primary goal of our proposed method is to optimize model parameters in a way that allows for efficient behavior on a new task with just one or a few gradient steps.

3.4 Evaluation mechanism

Rewards serve as the cornerstone of an agent's learning process in reinforcement learning, akin to labeled data in supervised learning. The effectiveness of an agent's learning through trial and error is heavily contingent upon the definition of the reward function. Hence, we formulate a reward function consisting of two primary components: 1) target location and 2) target distance from the tracker. In our reward function, if the target's position remains within the field of view, the reward increases by 1, whereas if the target strays out of the field of view, the reward decreases by 5. Additionally, a penalty term based on the distance between the target and the tracker is incorporated, compelling the tracker to progressively approach the target.

We employ the same evaluation metrics as DRLAVT [22] to validate the efficacy of our algorithm. Our primary objective is to ensure that the target remains within the field of view throughout the tracking process. Additionally, we aim for the tracker to achieve higher rewards.

$$\begin{aligned} AEL &= \frac{1}{N \times R} \sum_{n}^{N} \sum_{r}^{R} L_r^n \\ AER &= \frac{1}{N \times R} \sum_{n}^{N} \sum_{r}^{R} R_r^n \end{aligned} \tag{4}$$

Where L_r^n represents the Average Episode Length (AEL) of the nth tracking object at the rth epoch, and R_r^n represents the Average Episode Rewards (AER) of the nth tracking object at the rth epoch. Here, N is the total number of classes, and R is the total number of repeats. We set N = 5 and R = 20 for our experiments.

4 Experiments

Our evaluation of the experimental results aims to verify the Average Episode Length (AEL) and Average Episode Rewards (AER) of MRLAVT through the average of 20 repeated experiments. Higher AEL and AER values indicate better algorithm performance. All experiments were conducted on the laboratory server platform with Tesla P100 graphics cards.

Table 1 The tracking result Contrast of MRLAVT, DRLAVT, SiamRPN and KCF with different image input and output, adaption and no-adaption

Name	Input		Output			Metric		
	Color	RGBD	Force	Velocity	Position	AEL	AER	Avg Speeds
	–	✓	✓	–	–	60.72	−863.89	7194.13
Random	–	✓	–	✓	–	28.94	−969.64	6905.11
	–	✓	–	–	✓	128.40	−1083.20	7230.89
PBVS	–	✓	✓	–	–	409.10	1182.91	76.99
(SiamRPN)	–	✓	–	✓	–	679.56	2790.78	74.31
PBVS	–	✓	✓	–	–	366.58	904.32	91.11
(KCF)	–	✓	–	✓	–	591.06	245.40	87.02
DRLAVT	✓	–	–	–	✓	844.73	−1344.07	387.98
(ConvNet)	–	✓	–	–	✓	1001	14.8	419.20
MRLAVT	✓	–	–	–	✓	939.36	−1949.58	745.34
(ConvNet no adaption)	–	✓	–	–	✓	1001	495.06	756.57
MRLAVT	✓	–	–	–	✓	1001	346.29	514.97
(ConvNet with adaption)	–	✓	–	–	✓	1001	680.76	567.99
DRLAVT	✓	–	–	–	✓	955.23	−2584.86	222.62
(ResNet-18)	–	✓	–	–	✓	892.14	401.54	204.99
MRLAVT	✓	–	–	–	✓	852.95	−1544.87	203.75
(ResNet-18 no adaption)	–	✓	–	–	✓	1001	538.37	184.82
MRLAVT	✓	–	–	–	✓	1001	458.89	235.28
(ResNet-18 with adaption)	–	✓	–	–	✓	1001	381.74	190.68

The top two experimental results under each metrics are separately highlighted by red and green. The experimental results highlighted by blue correspond to Tables 3 and 4

Table 2 The tracking result contrast of MRLAVT and DRLAVT and SiamRPN with different perturbations

Name	Perturbations						Metric	
	Actuator noise	Time delay	Image blur				AEL	AER
			Level 1	Level 2	Level 3	Level 4		
	✓		–	–	–	–	220.18	39.73
	–	✓	–	–	–	–	423.67	1186.23
	–	–	✓	–	–	–	409.85	1182.91
PBVS	–	–	–	✓	–	–	390.48	1050.84
(SiamRPN)	–	–	–	–	✓	–	402.52	1108.77
	–	–	–	–	–	✓	393.74	1072.30
	✓	✓	–	–	✓	–	189.41	−83.419
	✓		–	–	–	–	1001	−88.05
	–	✓	–	–	–	–	1001	−43.39
	–	–	✓	–	–	–	1001	−305.36
DRLAVT	–	–	–	✓	–	–	1001	−711.90
(ConvNet)	–	–	–	–	✓	–	1001	−1522.77
	–	–	–	–	–	✓	1001	−2470.35
	✓	✓	–	–	✓	–	948.61	−1967.69
	✓		–	–	–	–	1001	502.63
	–	✓	–	–	–	–	1001	403.05
	–	–	✓	–	–	–	1001	−12.96
MRLAVT	–	–	–	✓	–	–	1001	−458.36
(ConvNet no adaption)	–	–	–	–	✓	–	1001	−978.54
	–	–	–	–	–	✓	1001	−1532.11
	✓	✓	–	–	✓	–	991	−1461.25
	✓		–	–	–	–	1001	690.80
	–	✓	–	–	–	–	1001	360.75
	–	–	✓	–	–	–	1001	354.32

Table 2 continued

Name	Perturbations						Metric	
	Actuator noise	Time delay	Image blur				AEL	AER
			Level 1	Level 2	Level 3	Level 4		
MRLAVT	–	–	–	✓	–	–	1001	278.82
(ConvNet With Adaption)	–	–	–	–	✓	–	1001	86.17
	–	–	–	–	–	✓	1001	10.60
	✓	✓	–	–	✓	–	1001	−360.62

The top two experimental results under each metrics are separately highlighted by red and green

4.1 Basic parameter setting

We conducted extensive experiments to validate the effectiveness of our algorithm, demonstrating its superiority through empirical results. Initially, we performed 300 training iterations in the simulation environment to enhance target tracking, with each episode lasting up to 1000 time steps. If the target was lost during tracking, the final experimental result would be significantly lower than 1000. Our Deep Q Network updated model parameters every tenth training iteration to refine the tracking strategy. In our reward function calculation, we applied a discount factor of 0.99, striking a balance between short-term and long-term rewards to ensure our DQN prioritizes both immediate gains and the ultimate tracking goal.

For optimization, we employed the classic Adam optimizer with a learning rate of $1e-5$ for both the outer and inner networks. Each task consisted of 5-way/6-shot samples, where 5-way refers to 5 categories of tracking targets and 6-shot refers to selecting 6 training data points from each category. These data points included the current state, actions, rewards, and new state. To ensure data richness and diversity, we randomly selected 5 categories from the pool of 18 target categories, along with 5 data points from each category. This approach ensured our tracker obtained a robust initial model capable of effectively adapting to various tasks with minimal adjustments.

The notation MRLAVT(1/2/3/5) indicates the number of inner layer model optimizations used in our algorithm, with the default parameter being 5 inner optimizations. The detailed results are presented in Tables 3 and 4. Interestingly, even with just 1, 2, or 3 inner optimizations, our algorithm outperforms DRLAVT, both with and without adaptation. Increasing the number of optimizations during initial model training in the inner layer leads to better-initialized models. Even with just a single optimization, significant improvements relative to DRLAVT are observed, as illustrated in Tables 3 and 4.

4.2 Experimental results

We validate the effectiveness and superiority of the MRLAVT algorithm through experiments. Firstly, we compare the results of MRLAVT with previous methods. Table 1 presents a comparison between MRLAVT, DRLAVT, and PBVS across different network structures and input data types. We assess the Average Episode Length (AEL) and Average Episode Reward (AER) under various tracking targets, using a 5-way/6-shot configuration. The results demonstrate that MRLAVT maintains continuous target tracking, with impressive performance even without adaptation. Furthermore, our algorithm exhibits superior performance, both in AEL and AER, compared to DRLAVT.

In Table 2, we contrast the performance of MRLAVT with ConvNet network structure against PBVS with SiamRPN network structure and DRLAVT with ConvNet network under different perturbations such as actor noise, time delay, and image blurring. The results indicate that MRLAVT achieves continuous target tracking with higher AER and average speed.

Additionally, Table 3 validates the performance of MRLAVT and DRLAVT on test tracking targets. The initial model demonstrates good performance without adaptation based on ConvNet and RGBD input. Upon adding a few adaptation steps, the experimental results show enhanced performance based on ConvNet and RGBD input, as depicted in Table 4.

Table 3 The evaluation results of each test object with no adaption of MRLAVT and DRLAVT based on convnet

	Targets	AEL	AER
DRLAVT	Asteroid 05	1001	275.32
	Asteroid 06	1001	286.12
	Capsule 03	1001	−216.13
	Rocket 03	1001	−88.48
	Satellite 03	1001	−445.67
	Station 03	1001	277.65
	Overall	1001	14.80
MRLAVT (1 times inner optimization)	Asteroid 05	1001	100.25
	Asteroid 06	1001	238.39
	Capsule 03	1001	198.48
	Rocket 03	1001	−221.42
	Satellite 03	1001	242.09
	Station 03	1001	−35.02
	Overall	1001	87.13
MRLAVT (2 times inner optimization)	Asteroid 05	1001	270.84
	Asteroid 06	1001	108.20
	Capsule 03	1001	635.70
	Rocket 03	1001	449.31
	Satellite 03	1001	401.80
	Station 03	1001	582.63
	Overall	1001	408.08
MRLAVT (3 times inner optimization)	Asteroid 05	1001	231.44
	Asteroid 06	1001	332.59
	Capsule 03	1001	8.76
	Rocket 03	1001	656.07
	Satellite 03	1001	677.82
	Station 03	1001	644.75
	Overall	1001	425.24
MRLAVT (5 times inner optimization (default))	Asteroid 05	1001	676.05
	Asteroid 06	1001	121.80
	Capsule 03	1001	344.30
	Rocket 03	1001	704.53
	Satellite 03	1001	678.77
	Station 03	1001	444.89
	Overall	1001	495.06

The top two experimental results under each metrics are separately highlighted by red and green. The MRLAVT(1/2/3/5) indicates the number of optimizations in inner layer model

Table 4 The evaluation results of each test object with adaption of MRLAVT and DRLAVT based on convnet

	Targets	AEL	AER
DRLAVT	Asteroid 05	1001	495.23
	Asteroid 06	1001	487.73
	Capsule 03	1001	116.62
	Rocket 03	1001	374.81
	Satellite 03	1001	445.67
	Station 03	1001	371.42
	Overall	1001	381.92
MRLAVT	Asteroid 05	1001	440.94
(1 times inner optimization)	Asteroid 06	1001	510.59
	Capsule 03	1001	580.64
	Rocket 03	1001	557.82
	Satellite 03	1001	611.22
	Station 03	1001	678.94
	Overall	1001	563.36
MRLAVT	Asteroid 05	1001	674.15
(2 times inner optimization)	Asteroid 06	1001	694.99
	Capsule 03	1001	636.89
	Rocket 03	1001	625.00
	Satellite 03	1001	698.03
	Station 03	1001	699.47
	Overall	1001	671.42
MRLAVT	Asteroid 05	1001	693.27
(3 times inner optimization)	Asteroid 06	1001	682.66
	Capsule 03	1001	665.98
	Rocket 03	1001	690.35
	Satellite 03	1001	699.85
	Station 03	1001	605.98
	Overall	1001	673.02
MRLAVT	Asteroid 05	1001	708.00
(5 times inner optimization(default))	Asteroid 06	1001	584.26
	Capsule 03	1001	684.47
	Rocket 03	1001	683.05
	Satellite 03	1001	703.70
	Station 03	1001	721.05
	Overall	1001	680.76

The top two experimental results under each metrics are separately highlighted by red and green. The MRLAVT(1/2/3/5) indicates the number of optimizations in inner layer model

Figures 5 and 6 display the active tracking trajectories of the MRLAVT algorithm and DRLAVT algorithm. It is evident that the MRLAVT algorithm tracks the target more effectively, exhibiting smaller fluctuations and ensuring a more stable target tracking.

Figures 7, 8 and 9 depict the error values of the tracking in three axes. A comparison with DRLAVT reveals that the MRLAVT algorithm achieves superior tracking performance, with smaller errors observed in all three dimensions.

Although our algorithm has higher tracking AEL, AER and generalization ability, the algorithm has higher computational complexity and requires longer computation time.

5 Conclusion

In this study, we introduce a novel method for active object tracking using meta-learning,

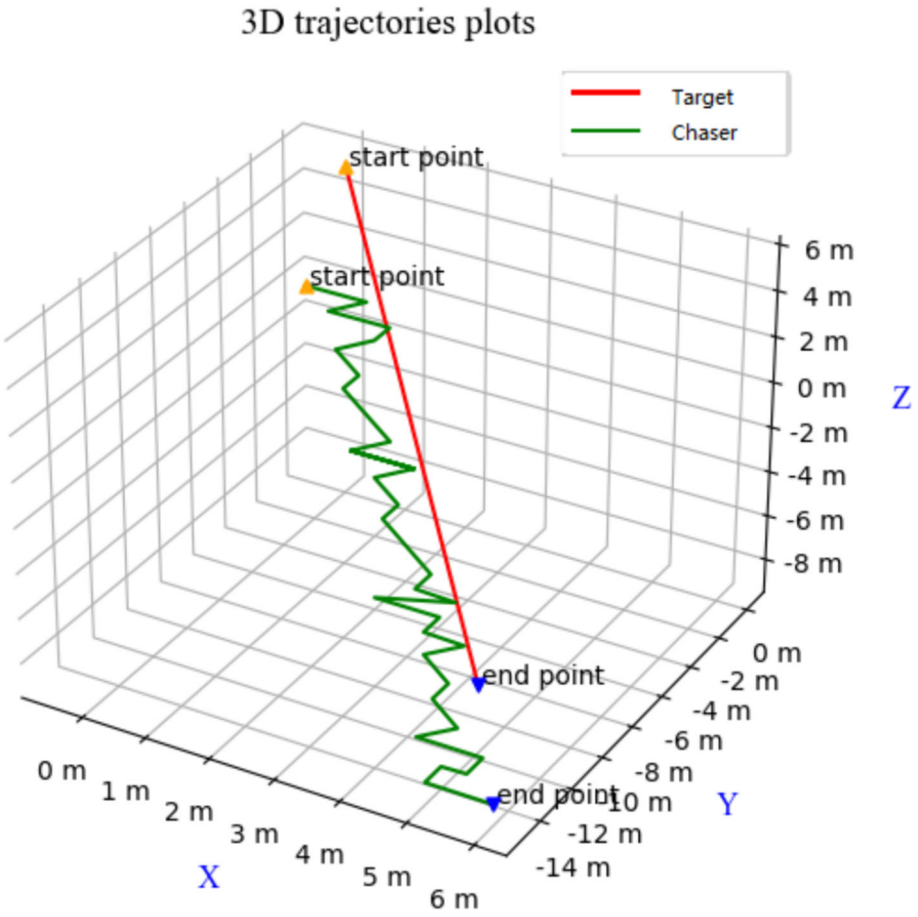

Fig. 5 The performance of DRLAVT about active tracking. The red line represents the moving trajectory of the target. The green line represents the tracker trajectory of our DRLAVT algorithm

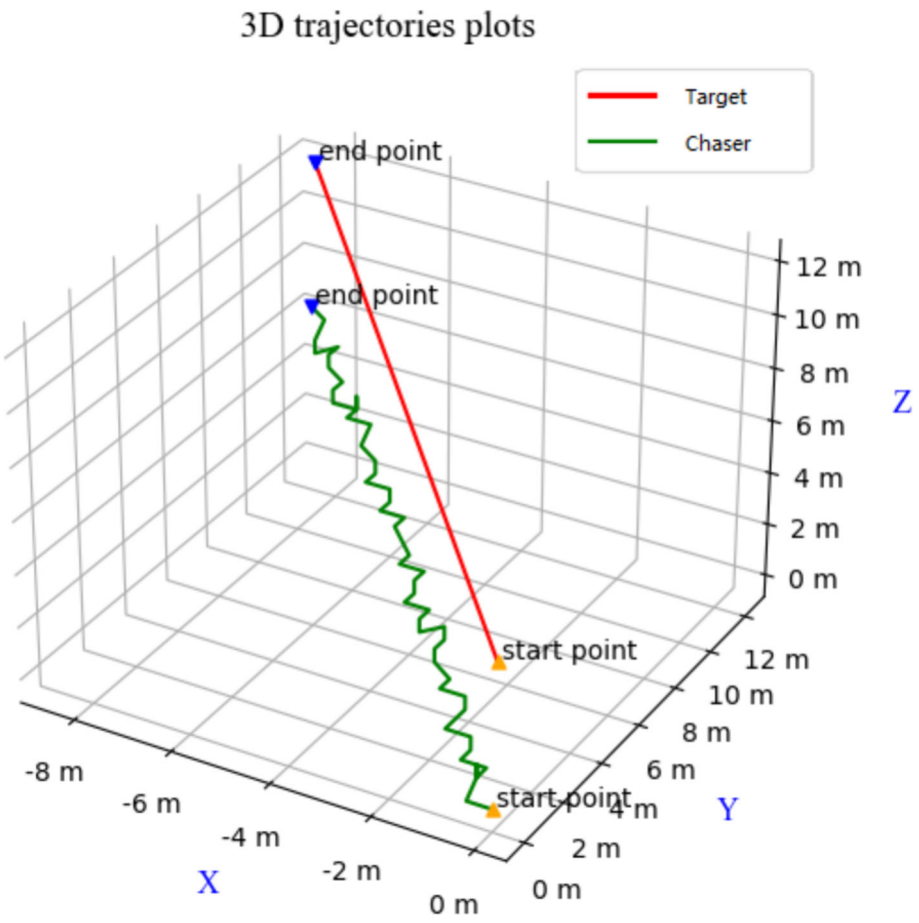

Fig. 6 The performance of MRLAVT about active tracking. The red line represents the moving trajectory of the target. The green line represents the tracker trajectory of our MRLAVT algorithm

termed MRLAVT. MRLAVT integrates target tracking and chaser control to achieve active object tracking, leveraging meta-learning for enhanced tracking performance. By quickly adapting to new tasks through accumulated experience, MRLAVT demonstrates strong generalization capabilities, reducing the reliance on extensive datasets. Even with limited tracking target datasets, MRLAVT can effectively train the model using alternative targets and generalize the learning. This approach ensures robust tracking performance while minimizing dataset dependency. Experimental results confirm the efficacy of MRLAVT in both generalized and non-generalized scenarios, underscoring its advanced nature. Notably, MRLAVT eliminates the need for training restart when tracking new targets, resulting in significant time savings without compromising tracking performance. In future, we will introduce lightweight object detection algorithms guarantee our algorithm has a wider range of practical applications.

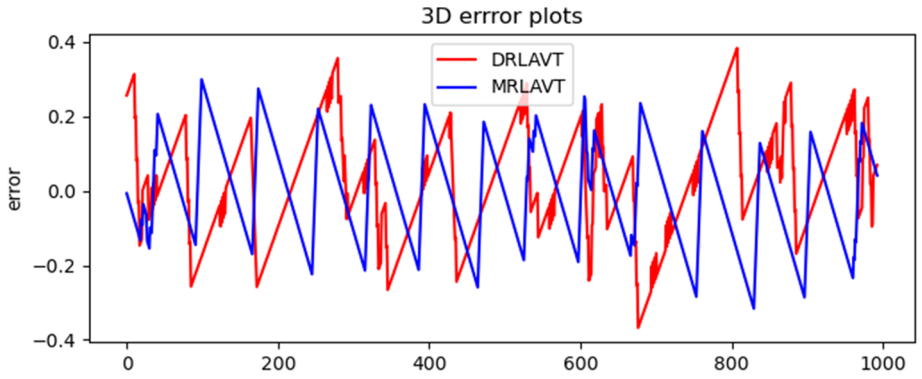

Fig. 7 The tracking error on the x-axis for MRLAVT and DRLAVT. The red line represents the tracking error of our DRLAVT. The green line represents the tracking error of our MRLAVT algorithm

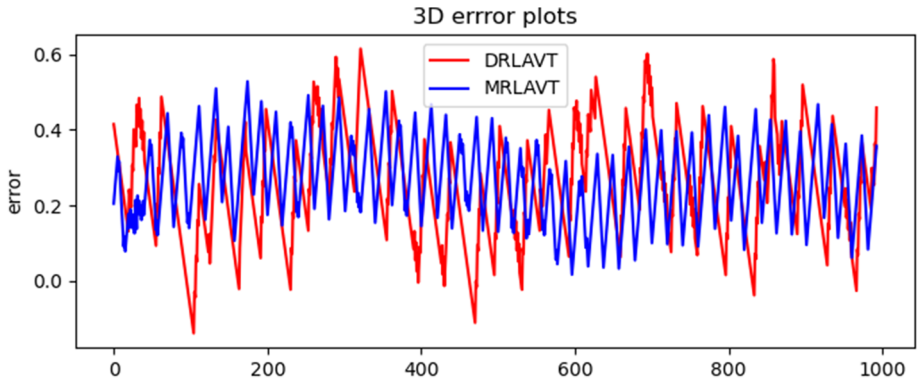

Fig. 8 tracking error on the y-axis for MRLAVT and DRLAVT. The red line represents the tracking error of our DRLAVT. The green line represents the tracking error of our MRLAVT algorithm

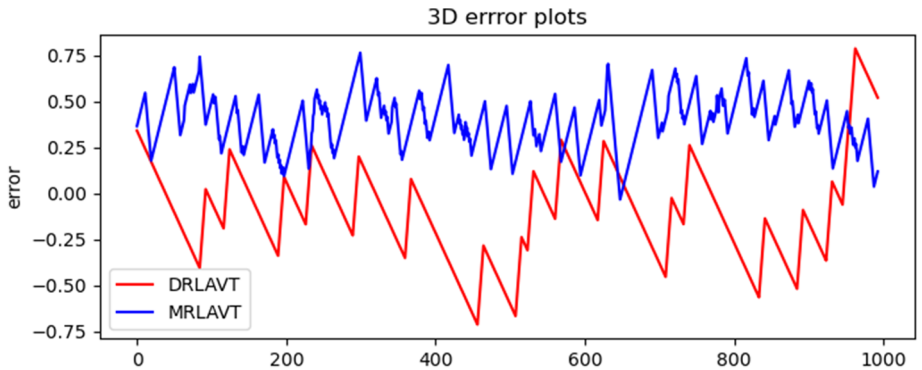

Fig. 9 tracking error on the z-axis for MRLAVT and DRLAVT. The red line represents the tracking error of our DRLAVT. The green line represents the tracking error of our MRLAVT algorithm

Data Availibility Statement The datasets generated for our paper can be requested from the corresponding author.

Declarations

Conflicts of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Zhou T, Porikli F, Crandall DJ, Van Gool L, Wang W (2022) A survey on deep learning technique for video segmentation. IEEE Trans Pattern Anal Mach Intell 45(6):7099–7122
2. Zhou T, Wang S, Zhou Y, Yao Y, Li J, Shao L (2020) Motion-attentive transition for zero-shot video object segmentation. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 13066–13073
3. Zhou T, Li J, Li X, Shao L (2021) Target-aware object discovery and association for unsupervised video multi-object segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6985–6994
4. Gu F, Lu J, Cai C, Zhu Q, Ju Z (2023) Eantrack: an efficient attention network for visual tracking. IEEE Trans Autom Sci Eng 1–18. https://doi.org/10.1109/TASE.2023.3319676
5. Gu F, Lu J, Cai C (2022) Rpformer: a robust parallel transformer for visual tracking in complex scenes. IEEE Trans Instrum Meas 71:1–14. https://doi.org/10.1109/TIM.2022.3170972
6. Gu F, Lu J, Cai C, Zhu Q, Ju Z (2024) Vtst: efficient visual tracking with a stereoscopic transformer. IEEE Trans Emerg Topics Comput Intell 8(3):2401–2416. https://doi.org/10.1109/TETCI.2024.3360303
7. Yuan D, Shu X, Liu Q, He Z (2023) Aligned spatial-temporal memory network for thermal infrared target tracking. IEEE Trans Circuits Syst II Express Briefs 70(3):1224–1228. https://doi.org/10.1109/TCSII.2022.3223871
8. Li J, Shi K, Xie G-S, Liu X, Zhang J, Zhou T (2024) Label-efficient few-shot semantic segmentation with unsupervised meta-training. In: Proceedings of the AAAI conference on artificial intelligence, vol 38, pp 3109–3117
9. Bouguet J-Y et al (2001) Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel corporation. 5(1–10):4
10. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. Image Vis Comput 21(1):99–110
11. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
12. Shan Y, Zhou X, Liu S, Zhang Y, Huang K (2020) Siamfpn: a deep learning method for accurate and real-time maritime ship tracking. IEEE Trans Circuits Syst Video Technol 31(1):315–325
13. Wang N, Song Y, Ma C, Zhou W, Liu W, Li H (2019) Unsupervised deep tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1308–1317
14. Li J, Xu J, Zhong F, Kong X, Qiao Y, Wang Y (2020) Pose-assisted multi-camera collaboration for active object tracking. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 759–766
15. Wang M, Qiu B, Zhu Z, Xue H, Zhou C (2021) Study on active tracking of underwater acoustic target based on deep convolution neural network. Appl Sci 11(16):7530
16. Luo W, Sun P, Zhong F, Liu W, Zhang T, Wang Y (2018) End-to-end active object tracking via reinforcement learning. In: International conference on machine learning, pp 3286–3295. PMLR
17. Luo W, Sun P, Zhong F, Liu W, Zhang T, Wang Y (2019) End-to-end active object tracking and its real-world deployment via reinforcement learning. IEEE Trans Pattern Anal Mach Intell 42(6):1317–1332
18. Xi M, Zhou Y, Chen Z, Zhou W, Li H (2021) Anti-distractor active object tracking in 3d environments. IEEE Trans Circuits Syst Video Technol 32(6):3697–3707
19. Tian L, Wu L, Wang Y, Yang G (2011) Binocular vision system design and its active object tracking. In: 2011 Fourth international symposium on computational intelligence and design, vol 1, pp 278–281. IEEE
20. Zhong F, Sun P, Luo W, Yan T, Wang Y (2021) Towards distraction-robust active visual tracking. In: International conference on machine learning, pp 12782–12792 . PMLR
21. Cortés-Pérez N, Torres-Méndez LA (2021) A mirror-based active vision system for underwater robots: from the design to active object tracking application. Frontiers in Robotics and AI 8
22. Zhou D, Sun G, Lei W, Wu L (2022) Space non-cooperative object active tracking with deep reinforcement learning. IEEE Trans Aerosp Electron Syst 1–15. https://doi.org/10.1109/TAES.2022.3211246

23. Yu ZL, Sun G (2023) A PID based meta-learning method about space non-cooperative active object tracking. IEEE Trans Veh Technol PP(8):1–12. https://doi.org/10.1109/TVT.2023.3279567
24. Yu Z (2023) An information fusion method for meta-tracker about online aerospace object tracking. J Intell Fuzzy Syst 1–13. https://doi.org/10.3233/jifs-230265
25. Yu Z, Lv J (2022) A fractional order method for meta-learning about aerospace target classification. IEEE Trans Aerosp Electron Syst 1–10. https://doi.org/10.1109/TAES.2022.3207448
26. Nachum O, Tang H, Lu X, Gu S, Lee H, Levine S (2019) Why does hierarchy (sometimes) work so well in reinforcement learning? arXiv preprint. arXiv:1909.10618
27. Pathak D, Agrawal P, Efros AA, Darrell T (2017) Curiosity-driven exploration by self-supervised prediction. In: International conference on machine learning, pp 2778–2787. PMLR
28. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
29. Supancic III J, Ramanan D (2017) Tracking as online decision-making: learning a policy from streaming videos with reinforcement learning. In: Proceedings of the IEEE international conference on computer vision, pp 322–331
30. Antoniou A, Edwards H, Storkey A (2018) How to train your maml. arXiv preprint arXiv:1810.09502
31. Yu Z, Sun G, Lv J (2022) A fractional-order momentum optimization approach of deep neural networks. Neural Comput Appl 34(9):7091–7111
32. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1199–1208
33. Koch G, Zemel R, Salakhutdinov R et al (2015) Siamese neural networks for one-shot image recognition. In: ICML Deep Learning Workshop, vol 2. Lille
34. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates, Inc., ???
35. Grant E, Finn C, Levine S, Darrell T, Griffiths T (2018) Recasting gradient-based meta-learning as hierarchical bayes. arXiv preprint arXiv:1801.08930
36. Mishra N, Rohaninejad M, Chen X, Abbeel P (2017) A simple neural attentive meta-learner. arXiv preprint arXiv:1707.03141
37. Munkhdalai T, Yu H (2017) Meta networks. In: International conference on machine learning, pp 2554–2563. PMLR
38. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T (2016) Meta-learning with memory-augmented neural networks. In: International conference on machine learning, pp 1842–1850. PMLR
39. Wang JX, Kurth-Nelson Z, Tirumala D, Soyer H, Leibo JZ, Munos R, Blundell C, Kumaran D, Botvinick, M (2016) Learning to reinforcement learn. arXiv preprint. arXiv:1611.05763
40. Yu Z, Lv J (2023) A pid controller method for meta-learning about aerospace target classification. Multimed Tools Appl. https://doi.org/10.1007/s11042-023-17022-0
41. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778