TS-VAT: Efficient Deployment of Teacher-Student Framework in Visual Active Tracking

Sicheng Jiang[1][0009−0004−4479−4145]

University of Sydney, Camperdown NSW 2050, AUS
sjia0275@uni.sydney.edu.au

Abstract. Applying deep reinforcement learning-based visual active tracking algorithms in real-world environments is challenging due to complex surface textures and lighting variations. To tackle these issues while enhancing training efficiency, we propose a teacher-student framework-based visual active tracking algorithm. Our algorithm trains the teacher module using deep reinforcement learning, followed by supervised training of the student module with labels generated by the trained teacher. Instead of rendering images, privileged information is leveraged to reduce the teacher module's state space, thereby accelerating the training process. To further optimize efficiency while training the student module, a student database is employed to prevent re-rendering images. Additionally, image segmentation and data augmentation are incorporated to enhance the robustness of the student module. Experimental results show that our approach outperforms comparative algorithms while substantially reducing computational resource usage and training time. Real-world deployments of the student module in a complex indoor environment demonstrate that our method exhibits strong adaptability.

Keywords: Deep Reinforcement Learning · Visual Active Tracking · Teacher-Student Framework.

1 Introduction

In recent years, applying deep reinforcement learning(DRL) for real-world deployment of Visual Active Tracking (VAT) has gained significant attention. [1][2][3][4]. Compared to passive visual tracking, which only focuses on the target objects within the frame, VAT can control actuators to change the position and direction of the image sensor, keeping the target object within the frame by maintaining an appropriate distance and angle with the target.

Among the input information that can complete the VAT task, the image information provided by a monocular camera has lower acquisition cost. However, real-world images often present high complexity to the varying lighting conditions and complex object surface textures, which significantly increase the difficulty of image analysis, thereby making the real-world deployment of DRL algorithms challenging [5].

In order to address the aforementioned issues, existing DRL approaches employ environment augmentation techniques to create simulation environment

that closely mimic real-world conditions [2][6][7]. The neural network is trained in the aforementioned environment, gaining experience in handling complex situations for application in real-world environments. However, these solutions significantly increase memory usage and face problems such as long training periods and debugging difficulties.

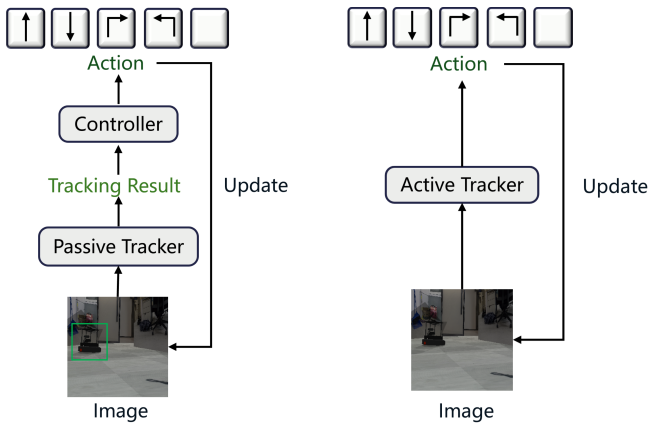
To address these issues, we have applied the Teacher-Student framework to the VAT field for the first time, proposing the Teacher-Student Framework-Based Visual Active Tracking Algorithm (TS-VAT).

Our TS-VAT, trained on an NVIDIA RTX 2070 Super, outperforms the comparative VAT algorithm trained on two RTX 3090 GPUs, reducing Video Random Access Memory(VRAM) usage by 92.5%, decreasing Random Access Memory(RAM) usage by 80%, shortening iteration time by 75%, and successfully completing tracking tasks in real-world environments.

2 Related Work

2.1 Visual Active Tracking

Visual Active Tracking algorithms can be categorized into end-to-end and non-end-to-end approaches (See Fig. 1). In non-end-to-end approaches, visual infor-



(a) Non-end-to-end approach (b) End-to-end approach

Fig. 1: Existing VAT approaches

mation is first processed by a passive tracker to identify the target, and the resulting passive tracking data is then input to the controller (Fig. 1a). This multi-stage structure has issues with error accumulation and difficulty in debugging. Whle the end-to-end approach uses an active tracker to directly output action signals from image inputs (Fig. 1b).

Many researchers have applied the end-to-end DRL algorithm to the field of VAT due to its high efficiency, adaptability, and lack of error accumulation. In 2020, Luo et al. [2] were the first to propose an end-to-end Visual Active Tracking algorithm, which outperformed the best passive trackers at the time. In 2021, Zhong et al. [4] proposed AD-VAT, a reinforcement learning method based on adversarial learning. In their algorithm, the tracker and target object are modeled as learnable entities that enhance their capabilities through interaction and competition during chasing and escaping. In 2023, Zhang et al. [3] introduced a binocular stereo matching-based VAT method, which enhances target identification accuracy by incorporating depth information. In 2024, Mao et al. [8] proposed CPO-AOT, an active object tracking framework based on constrained policy optimization, which combines asynchronous training mechanisms and a sparse reward function to address training challenges in high-dimensional state spaces and partially observable environments.

2.2 Teacher-student Framework

The Teacher-Student framework, also referred to as knowledge distillation, is a widely adopted technique for model compression and acceleration in deep learning. It was first introduced by Hinton et al. in 2015 [9].

In passive visual tracking field, Sohn et al. [10] were the first to propose a training method based on the Teacher-Student framework for semi-supervised learning in 2020, known as STAC. In 2022, Mi et al. [11] addressed the issue of the large data dependency in semi-supervised learning for passive visual tracking by proposing an active teacher method, achieving superior supervised performance with smaller labeling expenditures.

Given the efficiency of the Teacher-Student framework in computation and storage, applying it to reduce the training resource consumption of DRL-based Visual Active Tracking is a feasible and meaningful research direction.

3 Our Approach

In this section, we propose a TS-VAT algorithm. Section 3.1 introduces the modeling of tracking tasks in a virtual environment. Section 3.2 presents the specific design of TS-VAT, which requires training a teacher module and a student module, detailed respectively in Sections 3.3 and 3.4.

3.1 Problem Formulation

Simulation Environment Turtlebot4 is used as the robot platform for the experiment. The simulation environment for indoor tracking tasks utilizes Coppeliasim for robotic simulation, Pyrep for interfacing with reinforcement learning frameworks, and Tianshou [12] for managing reinforcement learning operations. The simulation environment is a 10m by 10m indoor space divided into three sections: a 5m by 10m area and two 5m by 5m areas. Rectangular obstacles representing furniture and black cylindrical distractors are placed in each area.

Tracking Target The tracking target is controlled by a designed algorithm based on the vector field histogram algorithm(Proposed by Borenstein et al. [13] in 1991.) that incorporates randomness, obstacle avoidance, and the ability to use obstacles to evade trackers.

Tracking Task Our tracking task is run frame by frame by the simulator, with the current simulation step defined as *step*. During the tracking process, the distance and angle between the tracking target and the tracker are defined as L and θ (see Fig. 3). The simulator executes actions at each *step* and updates L and θ. If either L or θ exceeds the threshold, or if a collision occurs, the tracking task is considered a failure.

3.2 TS-VAT Algorithm

To reduce the computational resource consumption of VAT, we propose the TS-VAT algorithm (See Fig. 2), which is carried out in three steps:

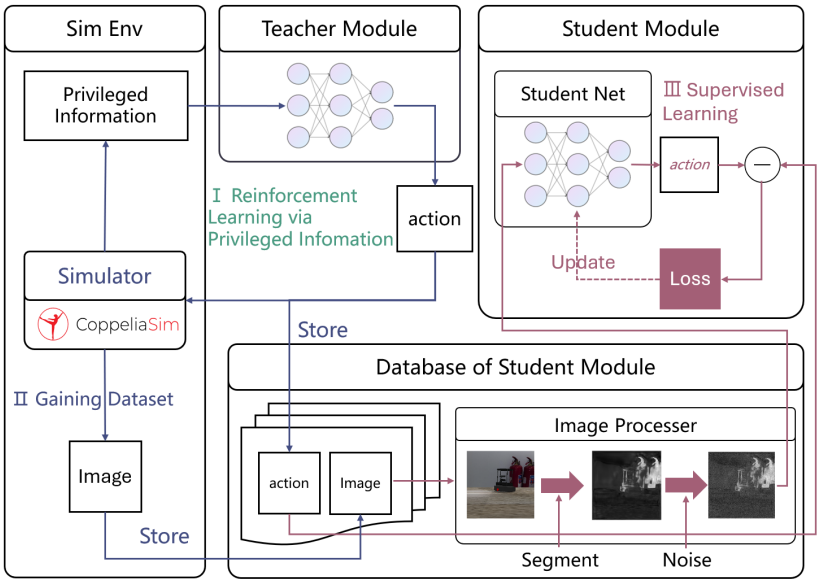


Fig. 2: TS-VAT algorithm

The first step involves training the Teacher Module using a deep reinforcement learning algorithm based on privileged information related to positioning and obstacle information. Since the dimension of privileged information is much smaller than that of image, VRAM consumption required for training is reduced.

The second step, as indicated by the blue line in Fig. 2, involves using the trained teacher module to interact with the simulator to obtain action-image

tuples, which are then stored in the student module's database. This step ensures that the simulator no longer needs to re-render the images when updating the image processor in step 3, thus improving the efficiency of the algorithm.

The third step, as indicated by the pink line in Fig. 2, involves using the image processor in the student module to perform image segmentation and data augmentation on the images in the database, thereby unifying the virtual-real image format and increase image complexity. Subsequently, based on the labels provided by the teacher module, a student module is trained to give the correct actions for the tracking task based on the processed images. This process enhances the robustness of the Student Module.

3.3 Teacher Module

Action space To facilitate the training of the teacher module, the discrete action space is set as the action space for the teacher module. This action space contains only five actions: move forward, move backward, rotate counterclockwise (turn left), rotate clockwise (turn right), and stop. Each action is defined as a, and the set of all actions is defined as $\mathcal{A}$.

State space To facilitate the training of the teacher module, the concept of privileged information is used to design the state space. Privileged information was proposed by Vanpik et al. [14], with the core idea of providing the network with global perspective hints during deep learning training to enhance the network's performance.

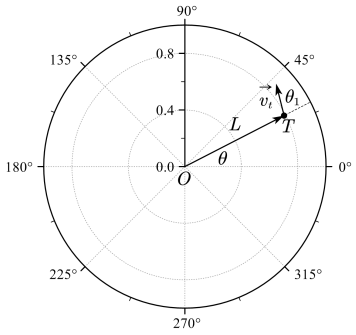


Fig. 3: tracking information

Fig. 3 represents the tracking situation at a certain moment. In the figure, the pole represents the position of the tracker, and the polar axis represents the orientation of the tracker. Point T and vector $\overrightarrow{v_t}$ represent the position of the tracking target and the current orientation of the tracker, respectively. L, θ, and θ_1 are the polar radius and polar angle of point T and the angle between $\overrightarrow{v_t}$ and $\overrightarrow{OT}$, respectively.

We consider the indoor tracking task as a 2D rigid body model. Since the tracking target has three degrees of freedom, its state can be determined by a set of L, θ, and θ_1. These three parameters, along with the obstacle sensors directly in front of the tracker, form the input for the teacher module. All possible values of these inputs constitute the state space of the teacher module.

Reward Function The reward function for the teacher module is defined as:

$$R_t = r_t + r_{1_t} + r_{o_t} \tag{1}$$

where r_t is the basic reward function, defined as:

$$r_t = A - \frac{|L_t - L_{best}|}{a} - \frac{|\theta_t - \theta_{best}|}{b} \tag{2}$$

r_{1_t} is the inducement reward function, defined as:

$$r_{1_t} = \begin{cases} 2 & r_t \geq 0.75 * A \\ 1.5 & r_t > r_{t-1} \text{ and } r_t < 0.75 * A \\ 0 & else \end{cases} \tag{3}$$

r_{o_t} is the obstacle avoidance reward function, defined as:

$$r_{o_t} = \sum_{i=1}^{7} G(O_i) \tag{4}$$

where $G(O_i)$ is the piecewise obstacle penalty function, defined as:

$$G(O_i) = \begin{cases} -0.4 & O_i < 0.2 \\ -0.1 & O_i < 0.4 \\ 0 & else \end{cases} \tag{5}$$

The above reward function decreases the reward when the tracking angle and position deviate from the optimal values or when obstacles are nearby. Additionally, a positive reward is given when the teacher module improves the tracking performance.

Module The teacher module employs the Dueling Double DQN(D3QN) [15][16] algorithm, which includes a Q-network and a target Q-network, both having the same structure.

To train the the module, first, the teacher module interacts with the environment using the ϵ-greedy strategy. The ϵ is set to 0.1. After the interaction is completed, the trajectory of the agent in a tracking task of length n is recorded as:

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n \tag{6}$$

This trajectory can be divided into n quintuples of the form (s_t, a_t, r_t, s_{t+1}), where s_t, a_t, and r_t are the privileged information, the selected action, and the reward obtained at time t, respectively.

After several interactions are completed, the temporal difference error is calculated as:

$$\delta_t = Q(s_t, a_t; \omega) - (r_t + \gamma \cdot \underset{a \in \mathcal{A}}{argmax}\, Q(s_{t+1}, a; \omega')) \tag{7}$$

where the function Q represents the forward propagation of network, and ω and ω' are the parameters of the Q-network and the target Q-network, respectively.

Finally, the gradient information of δ_t is backpropagated to update the network parameters.

3.4 Student Module

Student Database First, the teacher module is used to interact with the simulator using the same ϵ-greedy strategy with a larger $\epsilon = 0.2$. The trajectory of the teacher module, including image information during a tracking task of length n, is recorded as:

$$img_1, s_1, img_2, s_2, \cdots, img_n, s_n$$

where img_t and s_t are the image data from the unmanned platform camera and the privileged information of the teacher module at time t, respectively.

Subsequently, the array of tuples (IMG_t, T_t) are stored in the student module's database, where T_t is the action label output by the teacher module based on the privileged information s_t at time t, and IMG_t is the image obtained after processing img_t.

During testing in the virtual environment(section 4.3), the student module is trained using (img_t, T_t). In the real environment(section 4.4), the student module is trained using (IMG_t, T_t) to achieve virtual-to-real transfer.

Network The parameter update process for the student module is as follows: a tuple (img_t, T_t) is extracted from the training set of the student module's database. The student module performs forward propagation on img_t, obtaining the forward propagation result $logits_t$. Subsequently, the cross-entropy loss between $logits_t$ and the one-hot encoding of T_t is calculated:

$$H(logits_t, T_t) = -\log\left(\frac{e^{logits_t(T_t)}}{\sum\limits_{a \in \mathcal{A}} e^{\mathrm{logits}_t(a)}}\right) \tag{8}$$

Finally, the gradient information of $H(logits_t, T_t)$ is backpropagated, and the parameters of the student module are updated.

4 EXPERIMENTAL RESULTS

In this section, we detail the implementation of our approach and discuss the experimental campaign.

4.1 Experimental Setup

Teacher Module The structure of the Teacher Module is shown in Table 1 and its parameters are updated with Adam with learning rate $lr = 5e-5$ and batch size $N_b = 200$. The reward discount factor $\gamma = 0.9$ and buffer size $N_B = 1e5$. One simulation starts with $L = 1m$, $\theta = 0°$, $step = 0$, $collision = False$, and ends if $L \notin [L_{min}, L_{max}]$ or $\theta \notin [-\theta_{max}, \theta_{max}]$ or $step > step_{max}$ or $collision = True$, where $L_{min} = 0.4m$,$L_{max} = 2m$, $step_{max} = 1000$, $\theta_{max} = 30°$, $step$ is the number of simulation steps in the simulator, and $collision$ is a bool value indicating whether a collision has occurred.

Student Module student module's network structure is set to be SqueezeNET [17] and is trained with following arguments: $lr = 1e-4$, $N_b = 50$. One student module is tested in simulation environment on section 4.3 with its database's images not being processed. While The others are tested in real-world environment on section 4.4 with their databases' images being processed.

Computing Platform All teacher module and student modules are trained on a laptop equipped with NVIDIA RTX 2070 Super with 8GB of VRAM, an Intel Core processor i7-10750h (6 cores, 12 threads, 2.4 GHz base frequency) and 32 GB of DDR4 RAM.

Table 1: Network and GPU Models of comparative algorithm and teacher module

Algorithm	Layer					Params	GPU Model (NVIDIA RTX)
	1	2	3	4	5		
Comparative VAT algorithm	C8*8-16S4	C4*4-32S2	FC256	FC256	FC5 FC1	800k	2×3090
Teacher Module	FC128	FC128	FC32	FC32	FC5 FC1	20k	2070 Super

4.2 Comparison algorithms

Comparative VAT algorithm The comparative VAT algorithm employs the D3QN algorithm (see network structure in Table 1) with image input. It is trained using the same hyperparameters as TS-VAT on a workstation equipped with 2× NVIDIA RTX 3090 GPUs with 50GB of VRAM, an Intel Xeon 4210R processor (10 cores, 20 threads, 2.6 GHz base frequency), and 128GB of DDR4 RAM. This setup significantly surpasses the training platform of TS-VAT.

Artificial Potential Field Algorithm We design an artificial potential field method(Proposed by Khatib et al. [18] in 1985.) as a comparative algorithm, which controls the tracker by calculating the resultant force of the virtual repulsive force from obstacles and the virtual attractive force from the tracking target.

TS-VAT Using Image Augmentation We replace the image segmentation method to form a comparative TS-VAT algorithm. This approach is the same as the proposed TS-VAT except for its student database processor using Python scripts 'albumentations' to implement image augmentation.

4.3 VAT on Simulation Environment

We conduct reinforcement learning training for the teacher module and the comparative algorithms in the constructed virtual environment (See Fig. 4). After the training of the teacher module is completed, supervised learning training is conducted for the student module (See Fig. 6a). Three types of student modules are tested (See Table2). The experimental results show that the accuracy of the student module increases with the number of network parameters. After balancing the number of parameters and accuracy, SqueezeNet is ultimately set as the structure of the student module.

Table 2: Three pending network structures of student module

NetWork	Parameter	Depth	Accuracy
Comparative VAT Network (See Table 1)	0.8M	5 layers	81%
ResNET50 [19]	25M	50 layers	92%
SqueezeNET [17]	1.2M	18 layers	89%

The teacher module, student module, Comparative APF algorithm, and Comparative VAT algorithm are tested in the constructed environment. The testing process involves starting the tracking task from a starting position that the tracking unmanned platform has never encountered during training. Except for a maximum tracking step length of 200 and the tracker's forward speed and turning speed being 10% slower than the tracking target, the movement method of the tracking target and other termination conditions of the tracking task are the same as in the training process. The test is repeated 100 times, and the success rate and the ratio of the average test step length to the simulation step limit are recorded(See Table 3).

The results indicate that: (1) In terms of training effectiveness, compared to the comparative algorithms, the student module achieves a 6% increase in tracking success rate while reducing VRAM and RAM usage by 95.5% and 80%, respectively. With image input, the student module outperforms the APF algorithm, which uses privileged information, thereby demonstrating the effectiveness

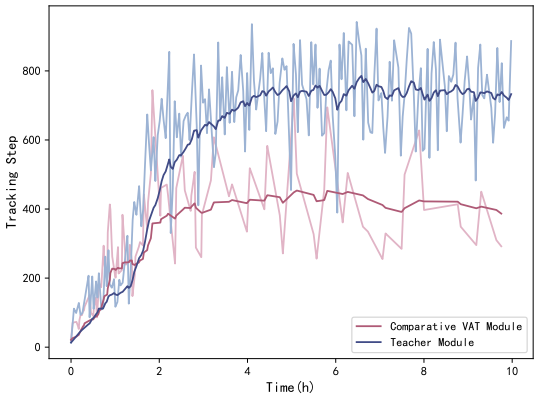


Fig. 4: Comparison of the Reinforcement Learning Training Processes of the Teacher Network and the Comparative VAT Algorithm

Table 3: Comparison of computing resources

TASK/Module	Train(Usage)			Test(Rate)	
	VRAM	RAM	Time	Success	Completion
Reinforcement Learning of Teacher Module	**<500MB**	**<10GB**	<6h	**91%**	**98.3%**
Collecting Database for Student Module	<7GB	<16GB	<30h	–	–
Supervised Learning of Student Module	<1800MB	<16GB	**<3h**	**87%**	**96.8%**
Reinforcement Learning of comparative VAT algorithm	>40GB	>80GB	<4h	81%	93.4%
APF Algorithm	–	–	–	84%	94.17%

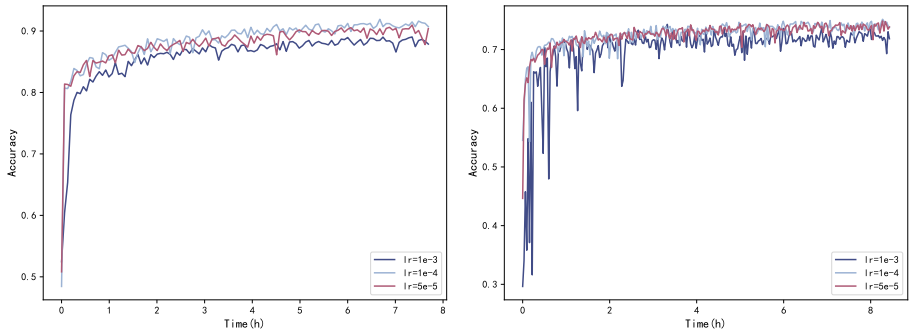
of our TS-VAT. However, the student module's tracking performance is inferior to that of the teacher module, suggesting that TS-VAT does not fully transfer the teacher's tracking strategy. (2) In terms of time cost, TS-VAT takes longer than the comparative algorithms, with the most significant time consumption attributed to the collection of the student module's training database. (3) TS-VAT, which runs on an NVIDIA RTX 2070 Super, outperforms the comparative VAT algorithm on two NVIDIA RTX 3090, highlighting its ability to reduce computational resource consumption.

4.4 VAT on Real-world Environment

To achieve virtual-to-real transfer, real images have been used as textures for basic environment augmentation, providing the tracking environment with more realistic textures(Fig. 5).

Fig. 5: Simulation environment with realistic textures

A new student module database is collected, and the images within it are processed before the new student module is retrained (See Fig. 6b). The results

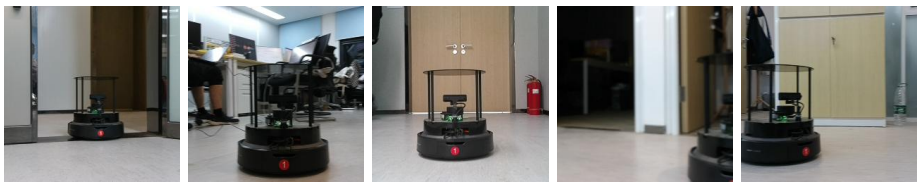


(a) Without image processing (b) With image processing

Fig. 6: The sim-to-real training process of the student network

show a decrease in the accuracy of the student module, indicating that the image processing method increases the diversity of the input images, helping the network to adapt to the complexity of real-world environments in advance. After the texture mapping and image processing operations are completed, the comparative VAT algorithm and the student module are trained. Subsequently a set of simple images(See Fig. 7) are used to test the tracking abilities of different networks(See Table 4).

Results in Table 4 indicate that due to the significant increase in image complexity, the training time of the comparative VAT algorithm lengthens considerably. Additionally, the test results indicate that the comparative VAT algorithm incorrectly provided control signals for following cases: right turn in Fig. 7a, left turn in Fig. 7b, and left turn in Fig. 7d. In these cases, the comparative VAT algorithm was disturbed by black objects in the background, leading to incorrect decisions, demonstrating its low adaptability to complex real-world scenarios.



(a) (b) (c) (d) (e)

Fig. 7: Real-world testing images

Table 4: Comparison of computing resources

TASK/Module	Train(Usage)			Test Accuracy
	VRAM	RAM	Time	
Supervised Learning of Student Module	**<3GB**	**<16GB**	**<3h**	**100%**
Reinforcement Learning of comparative VAT algorithm	>40GB	>80GB	>20h	40%

In contrast, the proposed TS-VAT algorithm performs very well in testing, demonstrating strong adaptability and anti-interference capability. Since the database has already been collected, the proposed TS-VAT algorithm can reduce Video Random Access Memory(VRAM) usage by 92.5%, decrease Random Access Memory(RAM) usage by 80%, shorten iteration time by 75%. This indicates that the proposed TS-VAT algorithm can significantly reduce computational resource consumption and greatly enhance algorithm iteration efficiency.

Additionally, compared to the training process in the virtual environment (See Table 3), the proposed TS-VAT's time consumption and computational resource usage are less sensitive to the complexity of tracking. This suggests that the proposed TS-VAT has the potential to be applied to more complex tasks.

For real-world deployment experiments, a relatively complex indoor office is selected as the deployment site. The site has a very complex background, including tables, chairs, clutter, computers, etc. Two obstacles, a chair and a trash can, representing common indoor obstacles, are placed within the site. The results of the student models for the two transfer methods, image segmentation and image augmentation, are shown in Fig. 8.

The experimental results indicate that the student modules from both transfer methods choose a diagonal movement when the tracking target makes a right-angle turn, moving directly in the direction of the tracking target's movement. This suggests that both have learned feasible tracking strategies. To quantitatively compare the differences between the two methods, a virtual scenario corresponding to the real tracking task is established, and ten repeated experiments are conducted in both real and virtual environments. After the experiments are

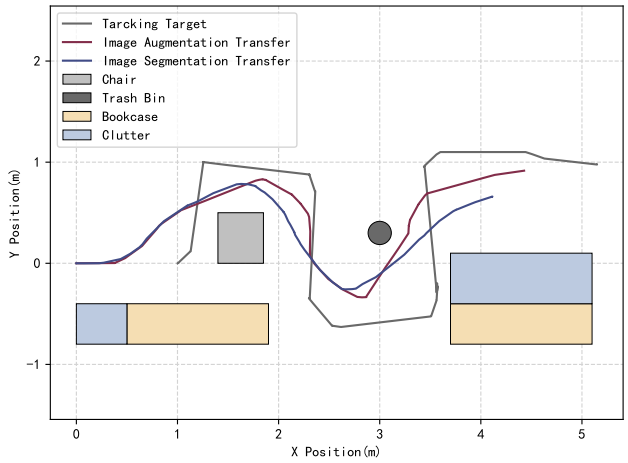


Fig. 8: Real-world tracking roadmap of student networks with two different transfer methods

completed, the following four metrics are recorded (See Table 5):

$$L_{abs} = \frac{1}{n}\sum_{t=0}^{n}|L_t - L_{best}| \tag{9}$$

$$L_{std} = \sqrt{\frac{1}{n}\sum_{t=0}^{n}(L_t - L_{best} - \mu_L)^2} \tag{10}$$

$$\theta_{abs} = \frac{1}{n}\sum_{t=0}^{n}|\theta_t - \theta_{best}| \tag{11}$$

$$\theta_{std} = \sqrt{\frac{1}{n}\sum_{t=0}^{n}(\theta_t - \theta_{best} - \mu_\theta)^2} \tag{12}$$

where L_t and θ_t are defined as shown in Fig.3, L_{Best} and θ_{Best} are the optimal tracking distance and optimal tracking angle, respectively, and μ_L and μ_θ are the average values of distance error and angle error, respectively.

The results in Table 5 show that in the virtual environment, there is no significant difference in performance between the image processing method and the image segmentation method. However, in the real environment, the performance of both methods decreases, with the image segmentation method showing a smaller decline and higher tracking success rate. This phenomenon suggests that the image segmentation method exhibits stronger adaptability due to its ability to unify the image formats between the simulation and real environments.

Table 5: Simlulation and real-world testing results of different TS-VAT algorithm

Environment	Transfer Method	L_{abs}	ΔL_{std}	θ_{abs}	θ_{std}	Success Rate
Simulation	Image Augmentation	$0.168m$	$0.075m$	2.312°	1.508°	100%
	Image Segmentation	$0.162m$	$0.072m$	1.219°	1.269°	100%
Real World	Image Augmentation	$0.249m$	$0.128m$	8.594°	11.541°	40%
	Image Segmentation	**0.146m**	**0.053m**	**5.183°**	**4.936°**	**70%**

5 CONCLUSIONS

In this paper, we propose TS-VAT, a teacher-student framework-based visual active tracking algorithm. By using privileged information and the teacher-student framework, we significantly reduce VRAM and RAM consumption and greatly enhance training efficiency. Through the incorporation of image segmentation and data augmentation, we improve the model's robustness, enabling its application in real-world environments. Experimental results show that TS-VAT achieves superior real-world tracking performance while using fewer computational resources and training time. Therefore, our research provides a valuable reference for future studies aiming to improve the training efficiency of deep reinforcement learning-based visual active tracking algorithms.

Acknowledgments. The author wishes to express his sincere thanks to Prof. Chen Chen and Mr. Nengwei Xu at the Beijing Institute of Technology for their invaluable and generous support.

References

1. Dionigi, A., Felicioni, S., Leomanni, M., Costante, G.: D-vat: End-to-end visual active tracking for micro aerial vehicles. IEEE Robotics and Automation Letters **9**(6), 5046–5053 (2024). https://doi.org/10.1109/LRA.2024.3385700
2. Luo, W., Sun, P., Zhong, F., Liu, W., Zhang, T., Wang, Y.: 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 (2020). https://doi.org/10.1109/TPAMI.2019.2899570
3. Zhang, B., Jin, S., Yang, S., Ouyang, Q., Zheng, Y., Shi, D.: End-to-end binocular active visual tracking based on reinforcement learning. In: 2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM). pp. 334–339 (2023). https://doi.org/10.1109/MLCCIM60412.2023.00054
4. Zhong, F., Sun, P., Luo, W., Yan, T., Wang, Y.: 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 (2021). https://doi.org/10.1109/TPAMI.2019.2952590
5. Devo, A., Dionigi, A., Costante, G.: Enhancing continuous control of mobile robots for end-to-end visual active tracking. Robotics and Autonomous Systems **142**, 103799 (Aug 2021). https://doi.org/10.1016/j.robot.2021.103799

6. Zhou, D., Sun, G., Lei, W., Wu, L.: Space Noncooperative Object Active Tracking With Deep Reinforcement Learning. IEEE Transactions on Aerospace and Electronic Systems **58**(6), 4902–4916 (Dec 2022). https://doi.org/10.1109/TAES.2022.3211246
7. Zhou, D., Sun, G., Zhang, Z., Wu, L.: On Deep Recurrent Reinforcement Learning for Active Visual Tracking of Space Noncooperative Objects. IEEE Robotics and Automation Letters **8**(8), 4418–4425 (Aug 2023). https://doi.org/10.1109/LRA.2023.3282792
8. Mao, T., Zheng, H., Zhai, X.B., Zhu, J.: Robust active target tracking via constrained policy optimization in partially observable environments. In: 2024 7th International Symposium on Autonomous Systems (ISAS). pp. 1–7 (2024). https://doi.org/10.1109/ISAS61044.2024.10552592
9. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015), https://arxiv.org/abs/1503.02531
10. Sohn, K., Zhang, Z., Li, C.L., Zhang, H., Lee, C.Y., Pfister, T.: A Simple Semi-Supervised Learning Framework for Object Detection. arXiv preprint arXiv:2005.04757 (2020). https://doi.org/10.48550/arXiv.2005.04757
11. Mi, P., Lin, J., Zhou, Y., Shen, Y., Luo, G., Sun, X., Cao, L., Fu, R., Xu, Q., Ji, R.: Active Teacher for Semi-Supervised Object Detection. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14482–14491 (2022)
12. Weng, J., Chen, H., Yan, D., You, K., Duburcq, A., Zhang, M., Su, Y., Su, H., Zhu, J.: Tianshou: A highly modularized deep reinforcement learning library. Journal of Machine Learning Research **23**(267), 1–6 (2022), http://jmlr.org/papers/v23/21-1127.html
13. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation **7**(3), 278–288 (Jun 1991). https://doi.org/10.1109/70.88137
14. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. Neural Networks **22**(5), 544–557 (Jul 2009). https://doi.org/10.1016/j.neunet.2009.06.042
15. van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. Proceedings of the AAAI Conference on Artificial Intelligence **30**(1) (Mar 2016). https://doi.org/10.1609/aaai.v30i1.10295
16. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling Network Architectures for Deep Reinforcement Learning. In: International conference on machine learning. pp. 1995–2003. PMLR (2016), https://proceedings.mlr.press/v48/wangf16.html
17. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018). https://doi.org/10.1109/CVPR.2018.00716
18. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. The international journal of robotics research **5**(1), 90–98 (1986). https://doi.org/10.1177/027836498600500106
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90