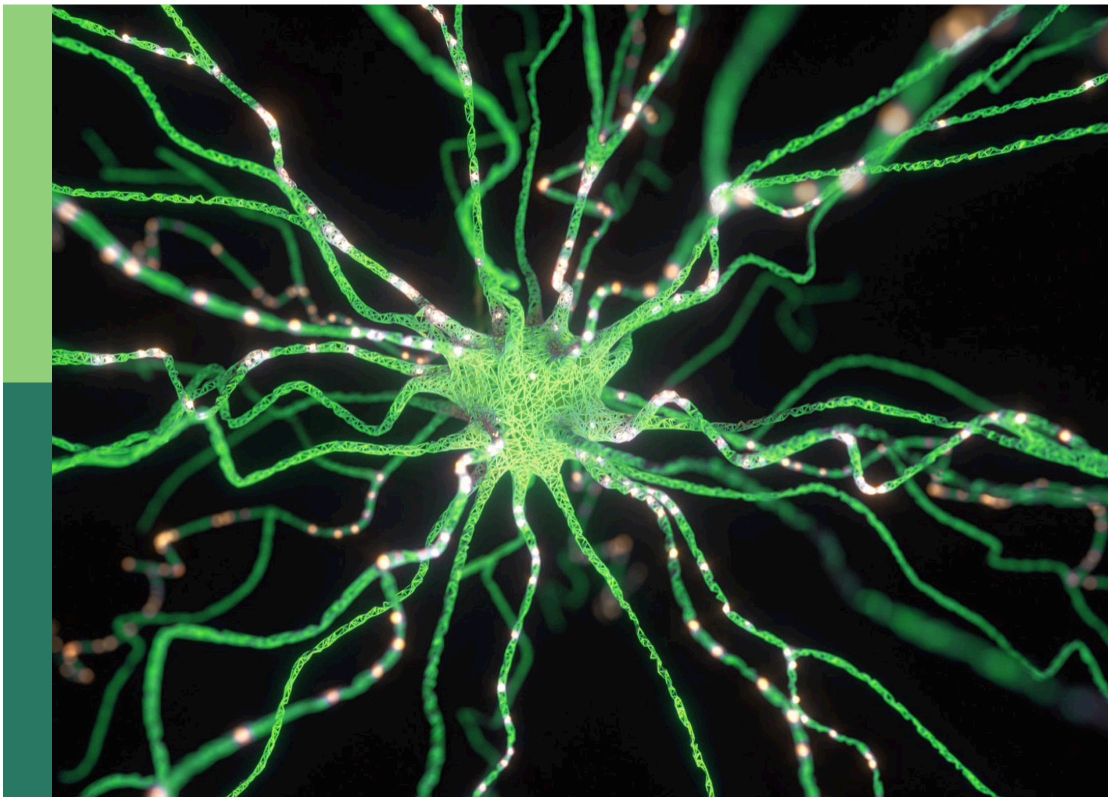
frontiers | Research Topics

Intelligent control and applications for robotics, volume II

Edited by
Yimin Zhou, Huiyu Zhou and Chen Qiao

Published in
Frontiers in Neurorobotics



ISSN 1664-8714
ISBN 978-2-8325-3925-5
DOI 10.3389/978-2-8325-3925-5

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Intelligent control and applications for robotics, volume II

Topic editors

Yimin Zhou — Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences (CAS), China
Huiyu Zhou — University of Leicester, United Kingdom
Chen Qiao — Xi'an Jiaotong University, China

Citation

Zhou, Y., Zhou, H., Qiao, C., eds. (2023). *Intelligent control and applications for robotics, volume II*. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-8325-3925-5

Table of contents

frontiers | Frontiers in Neurorobotics

TYPE Editorial
PUBLISHED 31 October 2023
DOI 10.3389/fnbot.2023.1282982

Check for updates

OPEN ACCESS

EDITED AND REVIEWED BY
Florian Röhrbein,
Technische Universität Chemnitz, Germany

*CORRESPONDENCE
Yimin Zhou
✉ ym.zhou@siat.ac.cn

RECEIVED 25 August 2023
ACCEPTED 23 October 2023
PUBLISHED 31 October 2023

CITATION
Zhou Y (2023) Editorial: Intelligent control and applications for robotics, volume II.
Front. Neurorobot. 17:1282982.
doi: 10.3389/fnbot.2023.1282982

Editorial: Intelligent control and applications for robotics, volume II

Yimin Zhou*

Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

KEYWORDS

robotics, perception, decision-making, autonomous control, applications

Editorial on the Research Topic
Intelligent control and applications for robotics, volume II

1. Introduction

Robotic technologies have undergone decades of the development and has now entered a highly intelligent stage, which is widely applied in various fields, including production and manufacturing, medical care, education, service Industries (Liu et al., 2020; Omisore et al., 2020). At present, the global robot market has exceeded $100 billion and is growing at a rate of over 17% annually. Among them, the Asia Pacific market is in an absolute leading position, with an estimated expenditure of 133 billion US dollars in 2020, accounting for 71% of the global market.

According to their functions and application fields, the global robot market can be divided into three categories: industrial robots, service robots, and special robots (Yang et al., 2021; Keroglou et al., 2023). Currently, the robotic technologies are being developed with the deep learning and artificial intelligence, where the deep learning technology can enable robots to obtain more accurate artificial intelligence so as to simulate human behaviors.

2. Analysis of the Research Topic

"Perception-Decision-Action" is the fundamental framework of the robots. In the perception phase, robots can perceive the environmental information via various sensors such as cameras, LiDAR, and inertial measurement units (IMUs). High-quality perception is crucial for the safe & efficient operation of the robots. The microelectromechanical system (MEMS) IMUs are widely used for the self-localization in the autonomous robots due to their small size and low power consumption. However, they are susceptible to random noise and bias errors, leading to lower measurement accuracy. Liu et al. conducted research on a low-cost MEMS IMU denoising method based on the deep learning. They proposed a hybrid denoising network which can combine the Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to eliminate the random noise in the raw data and calibrate IMU errors.

With the rapid advancement of the artificial intelligence technologies, vision has become one of the primary modalities for the robot perception (Yang et al., 2020). Robots can use image recognition techniques to extract positional and semantic information from the ambient environment for further decision-making. In practical application scenarios, complex and ambiguous background environments could lead to missed and false detections of small targets. Pei et al. improved the perceptual capability of YOLOv5 for small targets by enhancing input image resolution and retaining more feature information. In addition to the target identification and localization, in some scenarios, robots require to extract more complex semantic information from the visual data. Yang et al. utilized the human joint data, including joint positions, bone vectors, joint motion and bone motion data, to predict the human actions via a multi-scale attention spatiotemporal graph convolutional network. Just as humans can observe and identify objects from different perspectives, robots' active object recognition involves identifying targets through images captured from different viewpoints. Sun et al. developed a sampling strategy and training method for the viewpoint management during the robot perspective transformation process, which can help to determine the optimal planning for the active object recognition.

Planning and decision-making are crucial for the autonomy of the robot systems. The Monte Carlo Tree Search algorithm (MCTS) is a probabilistic search algorithm widely used in the decision-making and path planning problems. MCTS, due to its extensive random searches, is inherently inefficient when addressing individual problems. Li W. et al. introduced a self-learning MCTS (SL-MCTS) by combining MCTS with a dual-branch neural network. Compared with the traditional MCTS, the SL-MCTS is capable of finding better solutions with fewer iterations, significantly enhancing the search efficiency and quality of the path planning tasks. Zhang et al. applied the MCTS in the autonomous decision-making tasks in the aerial combat and incorporated deep reinforcement learning (DRL) to guide the MCTS searching for maneuvers in continuous action spaces, without relying on human knowledge to assist agents in the decision-making.

While DRL demonstrates outstanding performance in the planning and decision-making tasks due to its powerful self-learning capabilities, it might lead to a waste of computational resources when pursuing the maximization of long-term returns in atypical Markov decision processes. Additionally, errors in value function estimation can result in suboptimal policies. Pan et al. addressed these limitations in the atypical MDPs via the average reward method to form an unbiased, low-variance target Q-value with a simplified network architecture. Their approach showed significant advantages in terms of learning efficiency, effective control and computational resource utilization compared to the current methods. Meanwhile, Li S. et al. discussed the estimation bias issue, suggesting that a trade-off between the underestimation and overestimation can enhance DRL sample efficiency. They also introduced an Actor-Critic framework, which can learn values and policies within the same network and balances between the underestimation and overestimation.

In the case of multi-agent systems, besides individual autonomous decision-making, the coordinated actions among agents can improve the efficiency of the entire system. Hu et al. explored the problem of minimizing transportation costs in an automated guided vehicle cluster. They employed a hierarchical planning approach to decompose the integrated problem into an upper-level task allocation problem and a lower-level path planning problem. Hence the sum of the request cost and conflict delay cost of the entire system can be minimized via a hybrid discrete state transition algorithm based on the elite solution sets and a taboo list method.

In the execution phase, the robots heavily rely on the control algorithms to perform specific actions (Zheng et al., 2021). Tasks that involve extreme environmental conditions, high workloads and complex operational procedures demand the robots to have a particularly high-level of control precision. Zhao et al. have developed a variable damping controller for the end-effector of a space station robotic arm. With the reinforcement learning to control the variable damping of the robot limb, the resistance of the arm to the disturbance can be greatly enhanced.

3. Discussion and conclusion

With the continuous development of mobile robots, technologies such as multi-sensor fusion, control systems, and intelligent software are constantly being upgraded to meet the demands of more application scenarios. According to the statistics analysis, the mobile robot market will be expected to exceed $46 billion by 2025. The Mobile robots have been applied in various fields such as manufacturing, healthcare and military, with enormous potential and unlimited future development space, where the most cutting-edge technologies in the latest robotics field includes: flexible robot technology, liquid metal control technology, electromyography control technology, autonomous driving technology, virtual reality (VR) robot technology, photogenetic technology, brain computer interface (BCI) technology, machine learning (ML) technology, natural language processing (NLP) technology and blockchain technology. These technologies allows a wider range of application scenarios for the development of robots.

Overall, these technologies cover multiple aspects of the robotics field, from robot perception, decision-making to execution, from autonomous learning to interaction with humans, from single perception mode to multimodal perception, from hardware to software, from single decision-making to multi-task collaboration etc. These technologies have all driven the development of the robotics field. Some technologies have been fully developed, such as robot vision technology and robot grasping technology, while others are still rapidly developing, such as robot voice technology and robot navigation technology. Regardless of the technologies, they provide more potentials for the application of the robots and are constantly changing our way of life and work. At the same time, however, robots will also bring new challenges and issues, such as robot ethics, robot laws, robot safety, etc. These issues require us to jointly explore and solve to ensure the health and sustainable development of the robots.

Author contributions

YZ: Writing—original draft, Writing—review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was partially supported by the National Natural Science Foundation of China Refs. U1913201 and 61973296, in part by the Guangdong Basic and Applied Basic Research Foundation Ref. 2021B1515120038.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Keroglou, C., Kansizoglou, I., Michailidis, P., Oikonomou, K. M., Papapetros I. T., Dragkola, P., et al. (2023). A survey on technical challenges of assistive robotics for elder people in domestic environments: the ASPiDA concept. *IEEE Trans. Med. Robot. Bionics* 5, 196–205. doi: 10.1109/TMRB.2023.3261342

Liu, Y., Ma, X., Shu, L., Hancke, G. P., and Abu-Mahfouz, A. M. (2020). From Industry 4.0 to Agriculture 4.0: current status, enabling technologies, and research challenges. *IEEE Trans. Ind. Inform.* 17, 4322–4334. doi: 10.1109/TII.2020.3003910

Omisore, O. M., Han, S., Xiong, J., Li, H., Li, Z., and Wang, L. (2020). A review on flexible robotic systems for minimally invasive surgery. *IEEE Trans. Syst. Man Cybern. Syst.* 52, 631–644. doi: 10.1109/TSMC.2020.3026174

Yang, C., Zhu, Y., and Chen, Y. A. (2021). A review of human–machine cooperation in the robotics domain. *IEEE Trans. Hum. Mach. Syst.* 52, 12–25. doi: 10.1109/THMS.2021.3131684

Yang, J., Wang, C., Jiang, B., Song, H., and Meng, Q. (2020). Visual perception enabled industry intelligence: state of the art, challenges and prospects. *IEEE Trans. Ind. Inform.* 17, 2204–2219. doi: 10.1109/TII.2020.2998818

Zheng, Y., Xu, Z., and Wang, X. (2021). The fusion of deep learning and fuzzy systems: a state-of-the-art survey. *IEEE Trans. Fuzzy Syst.* 30, 2783–2799. doi: 10.1109/TFUZZ.2021.3062899

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 25 October 2022
DOI 10.3389/fnbot.2022.996412

OPEN ACCESS

EDITED BY
Chen Qiao,
Xi'an Jiaotong University, China

REVIEWED BY
Andreas Schweiger,
Airbus, Netherlands
Tongle Zhou,
Nanjing University of Aeronautics and Astronautics, China

*CORRESPONDENCE
Hongpeng Zhang
hphpzhang@126.com

RECEIVED 17 July 2022
ACCEPTED 05 October 2022
PUBLISHED 25 October 2022

CITATION
Zhang H, Zhou H, Wei Y and Huang C (2022) Autonomous maneuver decision-making method based on reinforcement learning and Monte Carlo tree search. *Front. Neurorobot.* 16:996412. doi: 10.3389/fnbot.2022.996412

Autonomous maneuver decision-making method based on reinforcement learning and Monte Carlo tree search

Hongpeng Zhang*, Huan Zhou, Yujie Wei and Changqiang Huang

Aeronautics Engineering College, Air Force Engineering University, Xi'an, China

Autonomous maneuver decision-making methods for air combat often rely on human knowledge, such as advantage functions, objective functions, or dense rewards in reinforcement learning, which limits the decision-making ability of unmanned combat aerial vehicle to the scope of human experience and result in slow progress in maneuver decision-making. Therefore, a maneuver decision-making method based on deep reinforcement learning and Monte Carlo tree search is proposed to investigate whether it is feasible for maneuver decision-making without human knowledge or advantage function. To this end, Monte Carlo tree search in continuous action space is proposed and neural networks-guided Monte Carlo tree search with self-play is utilized to improve the ability of air combat agents. It starts from random behaviors and generates samples consisting of states, actions, and results of air combat through self-play without using human knowledge. These samples are used to train the neural network, and the neural network with a greater winning rate is selected by simulations. Then, repeat the above process to gradually improve the maneuver decision-making ability. Simulations are conducted to verify the effectiveness of the proposed method, and the kinematic model of the missile is used in simulations instead of the missile engagement zone to test whether the maneuver decision-making method is effective or not. The simulation results of the fixed initial state and random initial state show that the proposed method is efficient and can meet the real-time requirement.

KEYWORDS

autonomous air combat, maneuver decision-making, deep reinforcement learning, Monte Carlo tree search, neural networks

Introduction

Autonomous air combat through unmanned combat aerial vehicles is the future of air combat and maneuver decision-making is the core of autonomous air combat. Therefore, it is urgent to build maneuver decision-making methods. Maneuver decision-making means that the aircraft chooses the appropriate maneuver (e.g., normal overload, tangential overload, and roll angle) to change its state according to the acquired information of the target (e.g., azimuth, velocity, height, and distance), so as to defeat the target.

Air combat can be divided into within-visual-range air combat and beyond-visual-range air combat. With the development of science and technology, the detection distance of airborne radar and the range of air-to-air missiles have been increased to hundreds of kilometers. Therefore, both sides of the air combat can discover each other and launch missiles at beyond-visual-range. Besides, the process of beyond-visual-range air combat is different from that of within-visual-range air combat because the principle and operation method between radar-guided missiles and infrared (IR) missiles are different. Radar-guided missiles are supposed to be used for beyond-visual-range and IR-guided missiles for within visual range, because the detection range of the radar is longer than that of the IR detector. The IR-guided missile does not need external equipment to provide target information after it is launched. It can obtain information about the target by means of its infrared detector and then attack the target. Therefore, the aircraft can retreat after launching missiles. However, after launching, there are two stages in the attack of radar-guided missiles, which are called the midcourse guidance stage and the terminal guidance stage. In the intermediate guidance stage, the radar of the missile is not activated. Thus, it is necessary for the aircraft radar to continuously detect the target, providing the information for the missile and guiding it to the target. During the terminal phase, the missile continues to chase the target according to the information provided by its radar until it hits the target or loses the target.

Therefore, the decision-making method in within-visual-range air combat cannot be used for beyond-visual-range air combat directly, so we need to find a new decision-making method for autonomous air combat. At the same time, the existing maneuver decision-making methods rely on human knowledge, which can also be regarded as a dense reward in reinforcement learning. Thus, sparse reward means only using the result of air combat (i.e., win or not), which does not rely on human knowledge. Moreover, if the task is complex, it is difficult to define and design human knowledge or dense reward. Therefore, it is necessary to explore maneuver decision-making methods using a sparse reward.

Recently, most of the research on maneuver decision-making is focused on within-visual-range air combat (Mcgrew et al., 2010; Guo et al., 2017; Du et al., 2018; Huang et al., 2018; Li et al., 2019). You et al. (2019) proposed a constrained parameter evolutionary learning algorithm for Bayesian network parameters learning with scarce data, which can be applied to unmanned aerial vehicle autonomous mission decision-making. Wu et al. (2011) proposed the situation assessment method of beyond-visual-range air combat based on missile attack area, and introduced a new angle advantage function, speed advantage function, and height advantage function into the situation assessment model. Li et al. (2020) proposed a cooperative occupation method for autonomous air combat of multiple UAVs based on weapon attack area. They used the weapon attack area and air combat geometric description for one-to-one air combat situation assessment and established a multiple UAVs cooperative occupation model based on the encircling advantage function. Therefore, the cooperative occupation problem was transformed into a mixed integer non-linear programming problem and solved by an improved discrete particle swarm optimization algorithm. However, the flight model in this study is two-dimensional, that is, the height of both sides of air combat is always the same in air combat, and the control quantities do not include roll angle, so this study can be further improved. Wei et al. (2015) proposed a cognitive control model with three-layer structure for multi-UAVs cooperative search according to the cognitive decision-making mode of humans performing searching behavior. The mission area is carried on cognitive match, reduction, and division based on this model and the fuzzy cluster idea. The simulation experiments indicate the great performance of the fuzzy cognitive decision-making method for cooperative search. Zhang et al. (2018) proposed a maneuver decision-making method based on the Q network and Nash equilibrium strategy, and combined the missile attack area in the reward function to improve the efficiency of reinforcement learning. However, the maneuver library of this method only contains five maneuvers, which cannot meet the needs of air combat. Hu et al. (2021) proposed to use the improved deep Q network (Mnih et al., 2015) for maneuver decisions in autonomous air combat, constructed the relative motion model, missile attack model, maneuver decision-making framework, designed the reward function for training agents, and replaced the strategy network in deep Q network with the perception situation layer and value fitting layer. This method improves the winning rate of air combat, but the maneuver library is relatively simple and difficult to meet the needs of air combat.

It is worth noting that deep reinforcement learning has achieved professional performance in video games (Watkins and Dayan, 1992; Hado et al., 2016; Matteo et al., 2017), board games such as GO (Silver et al., 2016, 2017; Schrittwieser et al., 2020), real-time strategy games such as StarCraft (Oriol et al., 2019), magnetic control of tokamak plasmas (Jonas et al., 2022), data fusion (Zhou et al., 2020b), and intention prediction of aerial targets under Uncertain and Incomplete Information (Zhou et al., 2020a). Therefore, using deep reinforcement learning to improve the level of air combat maneuver decision-making is a feasible direction. AlphaStar is a multi-agent reinforcement learning algorithm based on supervised learning. It introduces league training: three pools of agents (the main agents, the league exploiters, and the main exploiters), each initialized by supervised learning, were subsequently trained with reinforcement learning. In AlphaStar, each agent is initially trained through supervised learning on replays to imitate human actions. Concretely, it uses a dataset of 971,000 replays played on StarCraft II from the top 22% of players. Therefore, it can be concluded that two features of AlphaStar are multi-agent reinforcement learning and human knowledge. However, we mainly focus on one-on-one air combat, which means that a multi-agent algorithm is not suitable and we are supposed to use a single-agent algorithm to address this problem. Meanwhile, replays of games from top players are not difficult to obtain, but it is difficult and expensive to obtain data from human pilots, which means that we cannot use supervised learning as the first phase of AlphaStar.

Ma et al. (2020) described the cooperative occupation decision-making problem of multiple UAVs as a zero-sum matrix game problem, and proposed a solution of double oracle algorithm combined with neighborhood search. In maneuver decision-making, at first, the position to be occupied by each aircraft is determined, and then the target to be attacked by each aircraft is determined, to reduce the threat and increase the advantage. Yang et al. (2020) studied the evasive maneuver strategy of unmanned combat aircraft in BVR air combat, and the problem was solved by the hierarchical multi-objective evolutionary algorithm. In this method, the decision variables are classified according to the physical meanings and then coded independently. Four escape maneuvers are designed, including turning maneuver, vertical maneuver, horizontal maneuver, and terminal maneuver. The evolutionary algorithm is used to find approximate Pareto optimal solutions and reduce invalid solutions, thus, the efficiency of the algorithm is improved. Ma et al. (2018) built an air combat game environment and train the agent with deep Q-learning.

Eloy et al. (2020) studied the attack against static high-value targets in air combat. It analyzed the confrontation process with game theory and put forward a differential game method of air combat combined with the missile attack area (Wu and Nan, 2013; Li et al., 2015; Wang et al., 2019). In this method, the air combat process is divided into the attack stage and retreat stage, while the attacker is divided into leader and wingman. In the attack stage, the leader enters the target area and launches missiles, and the wingman flies in formation. In the retreat stage, the wingman protects the leader from the missile attack of the other party. However, the flight model of aircraft is two-dimensional rather than three-dimensional. However, the authenticity of the two-dimensional motion model is worse than that of the three-dimensional motion model, so the three-dimensional motion model should have been used. He et al. (2017) proposed a maneuver decision-making method based on Monte Carlo tree search (MCTS), and it uses MCTS to find the action with the greatest air combat advantage among the seven basic maneuvers. This method verifies the feasibility of MCTS in maneuver decision-making.

While human knowledge or dense reward can make the algorithm achieve the goal quickly, it also limits the diversity and potential of the algorithm to the scope of human experience. For example, AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, and AlphaZero can defeat the world champion without human knowledge and has found several joseki that human players have never found before. Meanwhile, AlphaGo with human knowledge was once defeated by the world champion Lee Sedol, but AlphaGo Zero without human knowledge has not been defeated by any human players ever since. Thus, it is a reasonable conjecture that human knowledge is not good enough for training purposes for autonomous weapon deployment, and we propose a method in this article for air combat to investigate whether it is feasible for maneuver decision-making without human knowledge.

To this end, an air combat maneuver decision-making method based on deep reinforcement learning and MCTS is proposed, which aims at investigating whether it is feasible for maneuver decision-making without human knowledge or dense reward. First, different from existing methods, this method does not use human knowledge to assist the agent in maneuvering decision-making, but only uses the outcome of air combat simulations. Second, existing methods often make maneuver decisions in discrete and finite action space (e.g., maneuver library consists of finite maneuvers), however, the proposed method is based on continuous action space, which is more reasonable than discrete action space. Third, to select actions in continuous space, we proposed the method of MCTS in continuous space which is different from MCTS of existing decision-making methods. Moreover, existing methods often use missile engagement zone in simulations, but the missile may miss the target even if the target is in the missile engagement zone, therefore, the kinematic model of the missile is used in simulations instead of missile engagement zone to test whether it can hit the target, which reflects whether the maneuver decision-making method is effective (Li, 2010; Zhang et al., 2015). Our research logic is: if it works well in simulations, we may consider investigating it in the real world and modifying it if it does not work well. However, if it does not work well even in simulations, we do not consider transferring it to the real world. Therefore, we do the first step here, that is, investigating the method in simulations to make sure it works well in simulations at least before transferring it to the real world.

The main contributions are as follows: (1) To investigate whether it is feasible for maneuver decision-making without human knowledge, we propose to use the algorithm of self-play and MCTS which learns to search actions in continuous action space. (2) We provide a method to address the problem of MCTS in continuous space since MCTS cannot be applied to continuous space directly. (3) The simulation results demonstrate that although maneuver decision-making without human knowledge cannot completely defeat that with human knowledge, it is still feasible in air combat. The rest of this paper is organized as follows: In Section Aircraft model and missile model, the motion dynamics model of aircraft and missile is established. In Section Maneuver decision-making method based on deep reinforcement learning and MCTS, the process of self-play and neural network training is described (Hinton and Salakhutdinov, 2006; Goodfellow et al., 2017), and the role of human knowledge in maneuver decision-making is interpreted.

In Section Experiments and results, the training results of the neural network and the simulation results of air combat are given, and the decision-making ability of the proposed method is discussed according to the simulation results. The method in this article is summarized in Section Conclusion.

Aircraft model and missile model

The aircraft model adopts normal overload, tangential overload, and roll angle as control parameters. To simplify the complexity of the problem, the angle of attack and the angle of side slip are regarded as zero and the ground coordinate system is treated as the inertial system, meanwhile, the effects of the rotation of the earth are overlooked. The kinematic and dynamic model is shown as follows (Williams, 1990):

$$\begin{cases} \dot{x} = v\cos\gamma\cos\psi \\ \dot{y} = v\cos\gamma\sin\psi \\ \dot{z} = v\sin\gamma \\ \dot{v} = g(n_x - \sin\gamma) \\ \dot{\gamma} = \frac{g}{v}(n_z\cos\mu - \cos\gamma) \\ \dot{\psi} = \frac{g}{v\cos\gamma}n_z\sin\mu \end{cases} \tag{1}$$

where x, y, and z indicate the positions of the aircraft in the inertial coordinate system; γ is the pitch angle, ψ is the yaw angle, v is the velocity, and g is the acceleration of gravity. Roll angle μ, tangential overload n_x, and normal overload n_z are control parameters. The kinematic model of the missile is Wang et al. (2019):

$$\begin{cases} \dot{x}_m = v_m\cos\gamma_m\cos\psi_m \\ \dot{y}_m = v_m\cos\gamma_m\sin\psi_m \\ \dot{z}_m = v_m\sin\gamma_m \end{cases} \tag{2}$$

where x_m, y_m, and z_m indicate the positions of the missile in the inertial coordinate system; v_m is the velocity, γ_m is the pitch angle, and ψ_m is the yaw angle. The dynamic model of the missile is:

$$\begin{cases} \dot{v}_m = \frac{(P_m - Q_m)g}{G_m} - g\sin\gamma_m \\ \dot{\psi}_m = \frac{n_{mc}g}{v_m\cos\gamma_m} \\ \dot{\gamma}_m = \frac{n_{mh}g}{v_m} - \frac{g\cos\gamma_m}{v_m} \end{cases} \tag{3}$$

where P_m and Q_m are thrust and air resistance, G_m is the mass of the missile, and n_{mc} and n_{mh} are control overload in the yaw direction and pitch direction. P_m, Q_m, and G_m can be calculated by the following formula (Fang et al., 2019):

$$P_m = \begin{cases} 12000 & t \le t_w \\ 0 & t > t_w \end{cases} \tag{4}$$

$$Q_m = \frac{1}{2}\rho v_m^2 S_m C_{Dm} \tag{5}$$

$$G_m = \begin{cases} 173.6 - 8.2t & t \le t_w \\ 108 & t > t_w \end{cases} \tag{6}$$

where $t_w = 8.0\text{s}$, $\rho = 0.607$, $S_m = 0.0324$, and $C_{Dm} = 0.9$. It is assumed that the guidance coefficient of proportional guidance law is K in control planes. The two overloads in yaw and pitch directions are defined as:

$$\begin{cases} n_{mc} = K \cdot \frac{v_m\cos\gamma_t}{g}\left[\dot{\beta} + \tan\varepsilon\tan(\varepsilon + \beta)\dot{\varepsilon}\right] \\ n_{mh} = \frac{v_m}{g}\frac{K}{\cos(\varepsilon+\beta)}\dot{\varepsilon} \end{cases} \tag{7}$$

$$\begin{cases} \beta = \arctan(r_y/r_x\ ) \\ \varepsilon = \arctan(r_z/\sqrt{r_x^2 + r_y^2}\ ) \end{cases} \tag{8}$$

$$\begin{cases} \dot{\beta} = (\dot{r}_y r_x - r_y\dot{r}_x)/(r_x^2 + r_y^2) \\ \dot{\varepsilon} = \frac{(r_x^2+r_y^2)\dot{r}_z - r_z(\dot{r}_x r_x + \dot{r}_y r_y)}{R^2\sqrt{r_x^2+r_y^2}} \end{cases} \tag{9}$$

where β and ε are yaw angle and pitch angle of the line of sight, and $\dot{\beta}$ and $\dot{\varepsilon}$ are the corresponding derivatives. The line of sight vector is the distance vector $\vec{r}$, where $r_x = x_t - x_m, r_y = y_t - y_m, r_z = z_t - z_m$ and $R = \|\vec{r}\| = \sqrt{r_x^2 + r_y^2 + r_z^2}$.

The maximum overload of the missile is 40. When the minimum distance between the missile and the target is <12 m, the target is regarded as a hit; when missile flight time exceeds 120 s and it still fails to hit the target, the target is regarded as missed; during the midcourse guidance stage, the target is regarded as missed when its azimuth relative to the aircraft exceeds 85°; during the final guidance stage, the target is regarded as missed when its azimuth relative to missile axis exceeds 70°.

Maneuver decision-making method based on deep reinforcement learning and MCTS

He et al. (2017) uses MCTS to find the maneuver that makes the most air combat advantage among the seven basic maneuvers, in which human knowledge is used to define the air combat advantage. However, its action space is discrete and only contains seven basic maneuvers. In this paper, the search scope of maneuver is extended from seven basic maneuvers to continuous action space, which contains countless maneuvers theoretically, and human knowledge is not used to assist maneuver decision-making, but only the outcome of air combat simulations. The main idea of the proposed reinforcement learning algorithm is to use neural networks to generate the maneuver and value in each state and then use the neural network-guided MCTS to search the maneuver in the continuous action space. The maneuver selected by MCTS is more effective than the maneuver directly generated by the neural network. Then, repeat the above steps in the self-play to generate training samples and update the neural network with these training samples to make the neural network more closely match the improved maneuver and self-play winner. The repetition steps are stopped and the training is regarded as good enough usually when the rating of the agent (Silver et al., 2016, 2017; Schrittwieser et al., 2020) or the scores obtained by the agent (Mnih et al., 2015; Hado et al., 2016) does not increase visibly. The new network is used in the next iteration to make MCTS more powerful.

AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, and AlphaZero can defeat the world champion without human knowledge and has found several joseki that human players have never found before. Meanwhile, AlphaGo with human knowledge was once defeated by the world champion Lee Sedol, but AlphaGo Zero without human knowledge has not been defeated by any human players ever since. Therefore, we write "While human knowledge or dense reward can make the algorithm achieve the goal quickly, it also limit the diversity and potential of the algorithm to the scope of human experience" in the introduction, which mainly refers to the game of GO but not the autonomous weapon deployment. However, it is a reasonable conjecture that human knowledge is not good enough for the training purposes for autonomous weapon deployment, thus we propose this method for air combat to investigate whether it is feasible for maneuver decision-making without human knowledge.

Our method is inspired by and built upon AlphaGo Zero. However, AlphaGo Zero is not suitable for air combat because of continuous action space, so we modified it to make it able to handle continuous action space. Since AlphaGo with human knowledge is defeated by AlphaGo Zero without human knowledge, we want to know if the method without human knowledge is feasible in air combat or even better than the method with human knowledge; therefore, we investigate the problem in this paper. It is true that human knowledge is indeed useful, and we will study maneuver methods with human knowledge in future. On the other hand, considering the development of AlphaGo, although the AlphaStar approach used human knowledge, a new approach called AlphaStar Zero may appear just like AlphaGo Zero, which can defeat AlphaStar and the world champion in the game of StarCraft II without using any human knowledge.

MCTS in continuous space

MCTS is usually used for searching in discrete action space (He et al., 2017; Silver et al., 2017; Hu et al., 2021). In this paper, we use neural networks to guide MCTS as in Silver et al. (2017). Since MCTS is typically used in discrete space and cannot be used in continuous space directly, we propose the method of MCTS in continuous space to address the problem of maneuver decision-making in air combat. The generation and selection of action in continuous space are shown in Figure 1.

The green rectangle in Figures 1, 2 is the continuous action space, which contains countless actions theoretically. Therefore, it cannot be searched by MCTS directly and we propose the following method to make MCTS able to search in continuous action space. First, a state is sent to neural networks as input and the neural network outputs the action and value according to the state, in which the action is regarded as the mean of a Gaussian distribution, the action output by the neural network is represented by the red rectangle in Figure 1. After that, a Gaussian distribution is acquired as shown in the blue shadow part in Figure 1. Then, N-1 actions are sampled from the Gaussian distribution, which are represented by the black rectangles in Figure 1, so N actions are acquired totally and MCTS is used to search for these N actions. Figure 2 illustrates the search process of MCTS in continuous space.

Each node s in the tree contains all actions of edges (s, a), and each edge stores a set $\{N(s, a), W(s, a), Q(s, a), P(s, a)\}$, where N represents the number of visits, W represents the total action value, Q represents the average action value, and P is the a priori probability of selecting this action, which can be computed by the Gaussian probability density function.

MCTS repeats four operations to find the action: selection, play, expansion, and backpropagation. Selection: take the current state as the root node, start the simulation from the root node, and stop until the simulation reaches the leaf node at time-step L. Before time-step L, the action is selected according to the a priori probability and average action value in the tree, $a_t =$ argmax $[Q(s_t, a)+U(s_t, a)]$ (Rosin, 2011),

$$U(s,a) = P(s,a)\frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)}$$

the probability of a_t is proportional to the maximum of $Q(s_t, a)+U(s_t, a)$, in which $Q(s_t,a_t) = W(s_t,a_t)/N(s_t,a_t)$. Here, $W(s_t,a_t)$ is computed by the value head of neural networks, and actions are generated by the acting head of neural networks, which is different from the original MCTS used in He et al. (2017) and Hu et al. (2021) since the original MCTS chooses action randomly instead of using neural networks.

Play: when in the selection step an action is chosen, which has not been stored in the tree, the play starts. Actions are selected in self-play until the leaf node s_L is reached, and the leaf node s_L means it has not been expanded.

Expansion: the neural network is used to evaluate the leaf node s_L added to the queue, expand the leaf node s_L, and each edge (s_L, a) is initialized to $\{N(s_L, a) = 0, W(s_L, a) = 0, Q(s_L, a) = 0, P(s_L, a) = p_a\}$ and p_a is the priori probability of the action. This part is another different part from the original MCTS used in He et al. (2017) and Hu et al. (2021), since the original MCTS evaluates the leaf node s_L by rollouts. However, the proposed method evaluates the leaf node s_L by neural networks, that is, the MCTS is guided by neural networks.

Backpropagation: update the number of visits and value of each step t in turn, $N(s_t,a_t) = N(s_t,a_t)+1$, $W(s_t,a_t) = W(s_t,a_t)+v$, $Q(s_t,a_t) = W(s_t,a_t)/N(s_t,a_t)$.

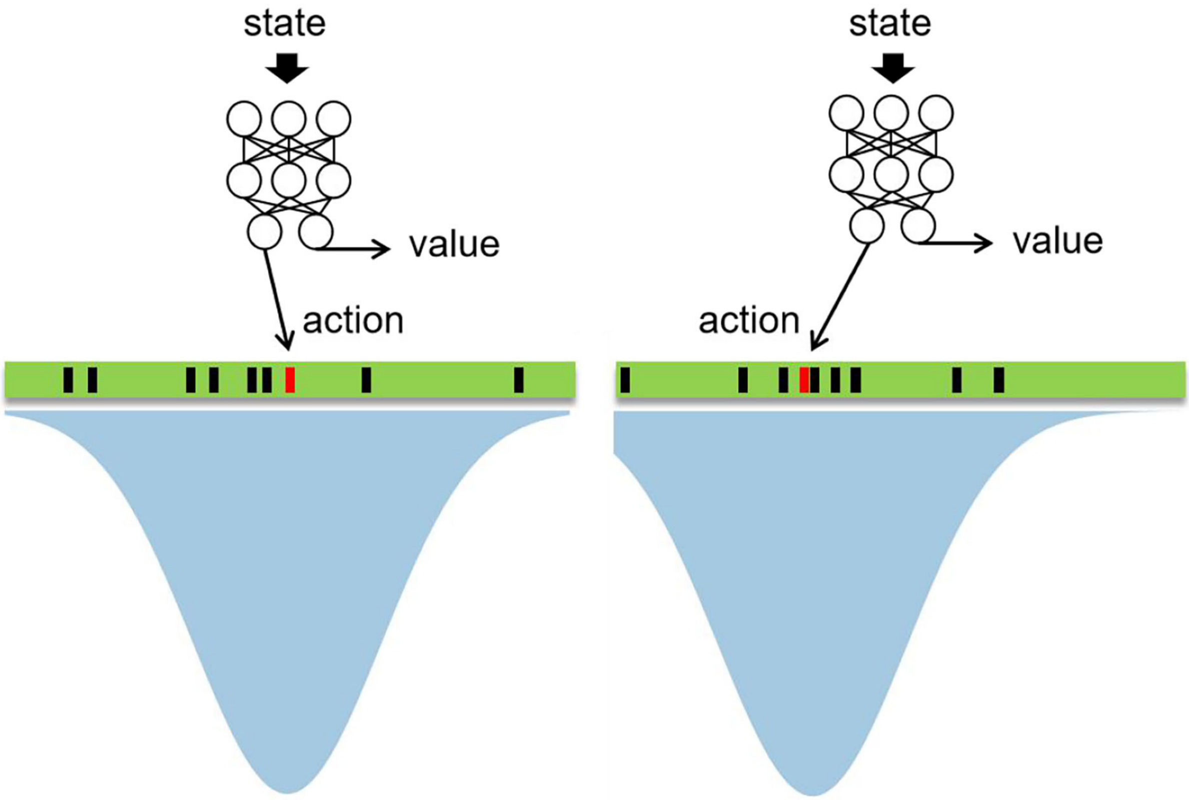


FIGURE 1
Generation and selection of action in continuous space.

After several iterations, MCTS outputs the action according to $a_t = \text{argmax}[Q(s_t, a)+U(s_t, a)]$ among N actions in continuous action space, as shown in the top right of Figure 2.

Reinforcement learning from self-play

Self-play reinforcement learning method has achieved professional performance in such games: chess (Baxter et al., 2000), othello (Sheppard, 2002), and poker (Moravcík, 2017). Therefore, this paper adopts self-play reinforcement learning for maneuver decision-making, and does not use any human knowledge. Starting from a completely random maneuver strategy, the neural network is trained by the data generated by self-play, so that the neural network can gradually produce effective maneuver strategies during the training pipeline. Figure 3 illustrates the self-play procedure.

As shown in Figure 3, at each time-step, the two sides of air combat execute the maneuvers selected by MCTS and reach the next time-step and a new state. In this state, the two sides continue to execute the maneuver obtained by MCTS until the final result of the simulation is obtained. The final result at the end T is $r_T = \{-1, 0, 1\}$, where -1 represents lose, 0 represents draw, and 1 represents win. It can be seen that there is no reward function in of self-play process except the final result of air combat, that is, human knowledge is not added to self-play, which is another feature of the proposed method. Self-play uses MCTS to generate state-action pairs in each iteration and takes these state-action pairs as samples to train the neural network. As shown in Figure 3, the air combat data of each time-step t is saved as (s_t, a_t, z_t) in the experience pool, $z_t = \pm r_T$ is the winner from the perspective of the current aircraft at time t. Uniform sampling (s_t, a_t, z_t) from all time-steps of the last iteration of self-play is used to train the network to minimize the error of prediction value and winner and the error of neural network output and MCTS output as shown in Figure 4, and the loss function is the sum of mean square error and L2 weight regularization.

To ensure the generalization ability of the neural network, the initial state of each game is randomly selected from the following scope: azimuth scope (-45°, 45°), speed scope (250, 400 m/s), and the distance between aircraft (40, 100 km). In self-play, MCTS is used to search 90 times for each decision. The first 10 maneuvers are sampled according to the visit count of each node and the subsequent maneuvers are those with the largest

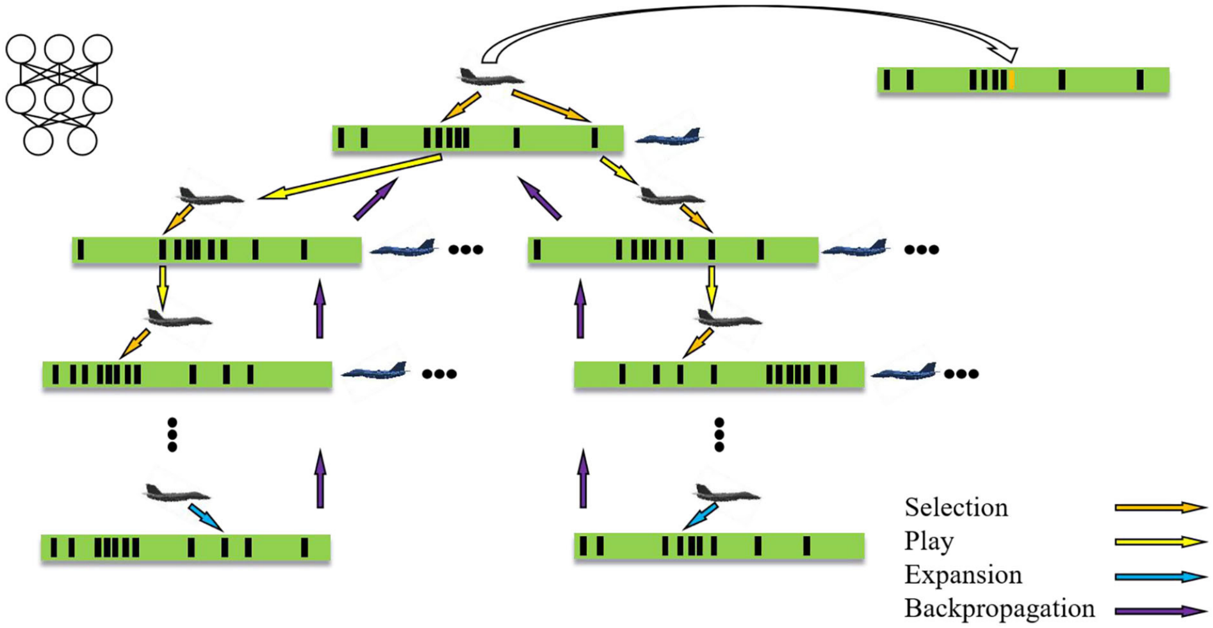


FIGURE 2
MCTS in continuous space.

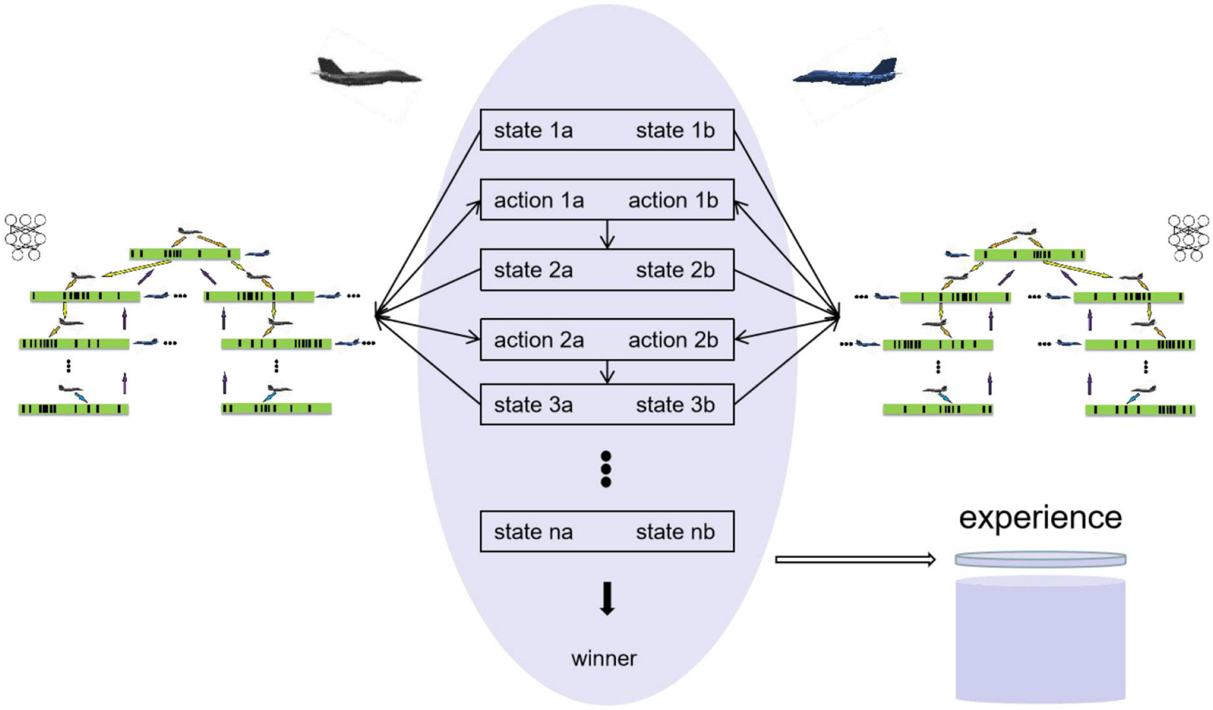


FIGURE 3
Air combat self-play.

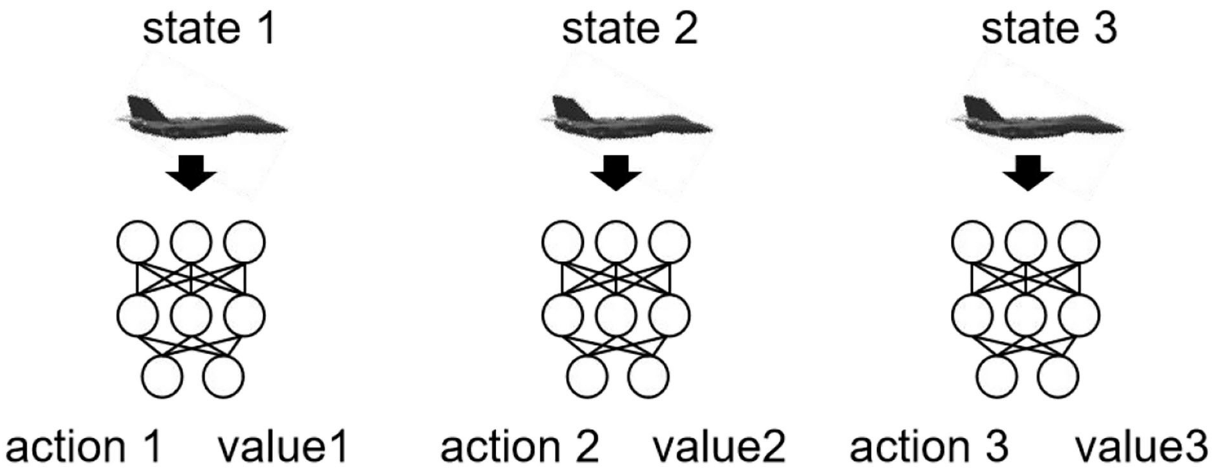


FIGURE 4
Training neural networks.

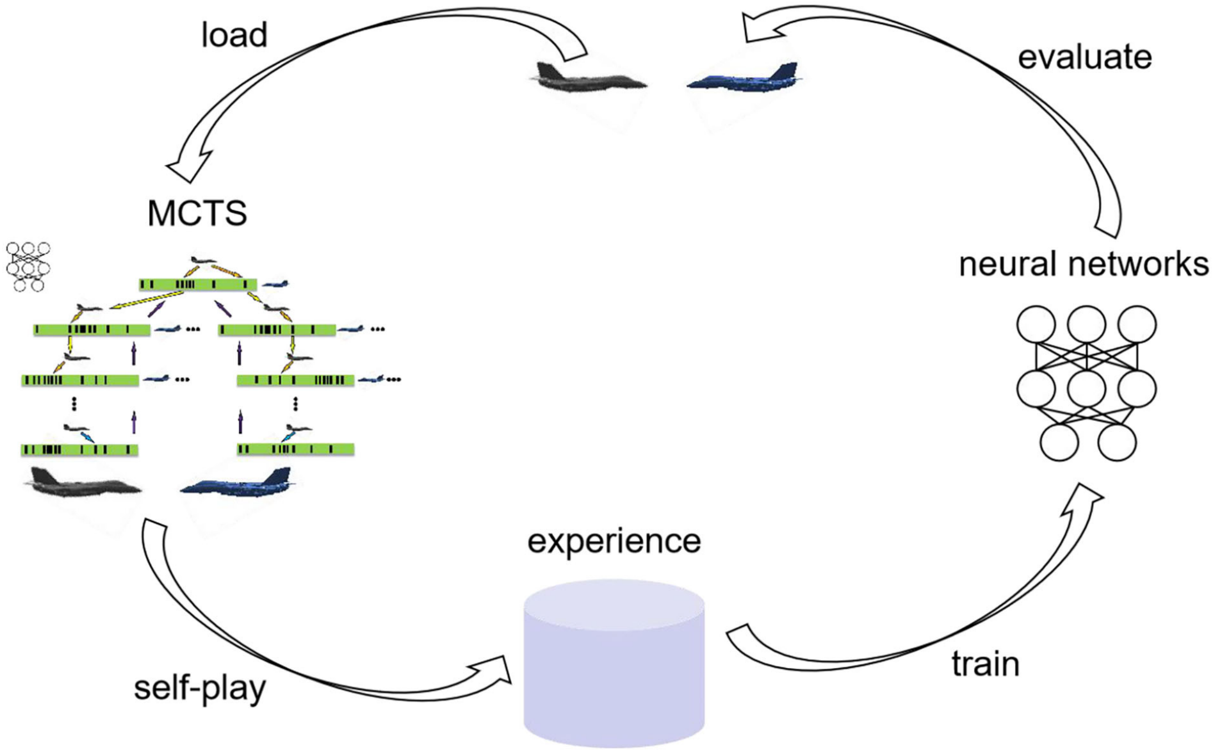


FIGURE 5
Training agent.

visit count, so as to balance the exploration and exploitation of the algorithm.

Figure 5 indicates the whole procedure of agent training. First, the agent generates air combat state-action pairs by MCTS in self-play and stores these data in the experience pool. Then, the neural network is trained with the data generated by 350 times of air combat self-play. During each training, 64 samples are uniformly sampled from the experience pool. The optimizer is stochastic gradient descent with a momentum of 0.9, and the L2 regularization coefficient is 0.0001. After 1,000 times of

```
Build neural networks with random weights
For iteration = 1,..., M do:
    Randomly initialize state $s_0$
    For t = 0,..., max step do
        State $s_t$
        Red side selects action by MCTS
        Blue side selects action by MCTS
        Simulate and reach the next state
        If find the winner:
            Beak
        Else:
            Continue
        t = t+1
    Store the state-action pairs and the winner
    If required amount of experience:
        Beak
    Else:
        Continue
Sample data from experience pool and train the neural networks
Save and evaluate the neural networks
If wins > 5 + failures:
    Load it as the current best neural network
```

Algorithm 1. Training agent.

training, a new neural network is obtained and saved. To ensure the quality of the data generated from self-play, the latest neural network after each training is evaluated: use the latest neural network to simulate air combat against the current best neural network 100 times. If the number of wins of the latest network is five more than failures, the latest neural network is loaded as the current best neural network and it is used to generate data in subsequent self-play, otherwise, the latest neural network is only saved but not loaded as the current best neural network. Algorithm 1 describes one iteration of agent training in Figure 5.

Air combat state and neural network architecture

The input of the neural network is a one-dimensional vector with 44 elements, which are composed of the state of the current time-step and the state of the first three time-steps. As shown in Table 1, each state contains 11 quantities: $\psi, \gamma, v, z, d, f_1, \psi_1, \gamma_1, d_1, \beta, f_2$, where ψ and γ are yaw angle and pitch angle of velocity vector relative to the line of sight, v is the velocity of the aircraft, z is the flight altitude, d is the distance between the two sides in air combat, and r_1 and r_2 are the coordinates of the two sides, respectively. f_1 represents whether our side launched a missile. Where ψ_1 and γ_1 are yaw angle and pitch angle of the missile's velocity vector relative to line of sight. d_1 is the distance between the missile and the other side and r_{m1} is the coordinate of the missile of the side in air combat. β is heading crossing angle, that is the angle between two velocity vectors of the two sides, which is represented by v_1 and v_2 in Table 1. f_2 represents whether the other side launched missile. The input layer is followed by three hidden layers. The number of neurons of the first two layers is 128 and the number of neurons of the third layer is 64. Finally, it output five quantities. The first three outputs are normal overload, tangential overload, and roll angle, respectively. The fourth output is whether to launch the missile and the fifth output is the value of the current state. The activation function is tanh.

TABLE 1 Air combat state.

State	Symbol	Formula
Yaw angle	ψ	$\psi = \psi + \int \frac{g}{v\cos\gamma} n_z \sin\mu \, \mathrm{dt}$
Pitch angle	γ	$\gamma = \gamma_0 + \int \frac{g}{v}(n_z \cos\mu - \cos\gamma) \, \mathrm{dt}$
Velocity	v	$v = v_0 + \int g(n_x - \sin\gamma) \, \mathrm{dt}$
Altitude	z	$z = z_0 + \int v \sin\gamma \, \mathrm{dt}$
Distance between the two sides	d	$d = \|r_1 - r_2\|$
Launch missile	f_1	0 or 1
Yaw angle of missile	ψ_1	$\psi_m = \psi_{m0} + \int \frac{n_{mc} g}{v_m \cos\gamma_m} \mathrm{dt}$
Pitch angle of missile	γ_1	$\gamma_m = \gamma_{m0} + \int \frac{n_{mh} g}{v_m} - \frac{g\cos\gamma_m}{v_m} \, \mathrm{dt}$
Distance between the missile and the other side	d_1	$d = \|r_{m1} - r_2\|$
Heading crossing angle	β	$\beta = \arccos(\frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|})$
Launch missile of the other side	f_2	0 or 1

Experiments and results

Parameter setting

The maximum flight speed is 420 m/s, and the minimum flight speed is 90 m/s; The maximum flight altitude is 20,000 m and the minimum flight altitude is 50 m; the initial roll angle is always zero; the decision interval is 1 s and the maximum simulation time is 200 s. The outcome of the air combat simulation is defined as follows: if the missile hit the target, record it as a win; either the aircraft or the missile misses the target, it is regarded as missing the target; when the flight altitude of one side is greater than the maximum altitude or less than the minimum altitude, if the other side has launched missile and does not miss the target, record it as lose, otherwise, record it as a draw; when both sides miss the target, record it as a draw. The decision interval is 1 s, because it is common in the field since previous work (Guo et al., 2017; Du et al., 2018; Huang et al., 2018) usually uses the decision interval of 1 s. Meanwhile, a shorter decision interval requires more computational sources and a longer time span is obviously irrational.

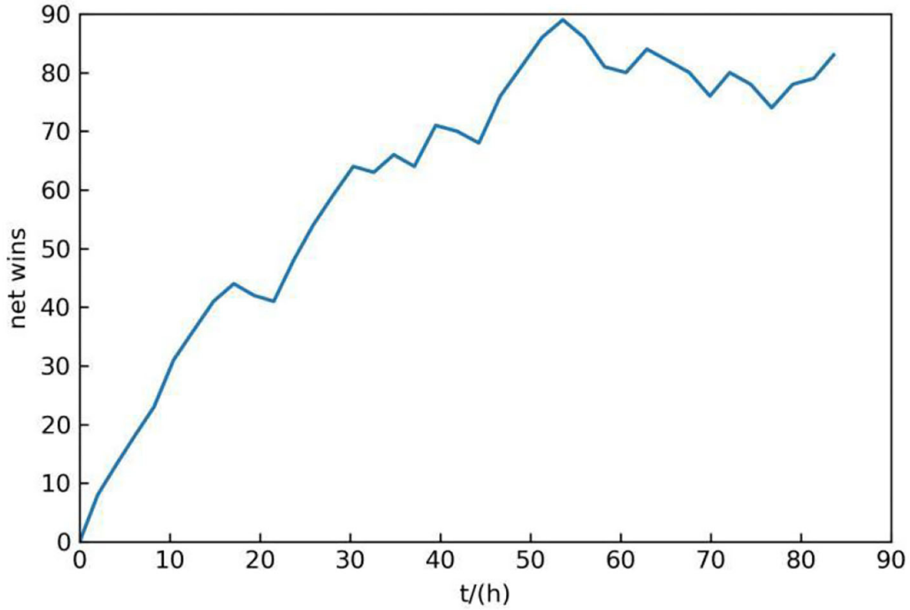


FIGURE 6
Neural networks training result.

TABLE 2 Statistic results.

Initial state	Win	Lose	Draw	Average time(s)
Fixed	23	17	60	0.38
Random	22	21	57	0.37

It is true that maneuver decisions are not of any use if decisions cannot be done in a reasonable time span. Here, the rate for maneuver decision of 1 per 1 s does not mean that the maneuver is static within 1 s. For example, the aircraft takes the maneuver of changing the roll angle from 0 degrees to 45 degrees within 1 s (the case of 1 maneuver per second), thus, it gradually increases its roll angle from 0 to 45 degrees, which is a dynamic process. On the other hand, increasing the roll angle from 0 to 45 degrees may be interpreted as three maneuvers as well, for example, 0–15, 15–25, and 25–45 degrees. More importantly, even if we send several different maneuvers to the real aircraft within 1 s (such as changing the roll angle from 0 to 30 degrees, then changing it from 30 to −10 degrees, and then changing it from −10 to 50 degrees), it may not be able to realize it because of the limitations of the hardware (e.g., aircraft servomechanism). On the other hand, even if the real aircraft can realize it, it is unacceptable, because it is harmful to the aircraft to change its maneuver several times within 1 s (lack of aircraft strength). Minimum reaction time of the human brain is ∼0.1 s. Meanwhile, it takes much more than 0.1 s for a human to decide what to do before the reaction, namely decision-making time. Therefore, the time span of 1 s is appropriate for a real-world application.

Results and analysis

Neural networks training result

In the process of self-play, record the net number of wins of each latest neural network in 100 times air combat, that is, subtract the number of failures from the number of wins. The reason why 100 times is selected is that 100 times is enough to distinguish the better one from both sides of the competition and does not cause too much time consumption. The total training time is about 84 h, and the change of net wins with time is shown in Figure 6.

As can be seen from Figure 6, the number of net wins is increasing along with the training. Although it sometimes decreases in the training process, it generally shows an upward trend, which indicates that the maneuvering decision-making ability of the proposed method gradually becomes more effective during self-play.

Air combat simulation results

We verify the effectiveness of the method we proposed by a fight against the MCTS method (He et al., 2017): (1) 100 simulations with a fixed initial state of the following simulation

2, which is a fair initial state for both sides. (2) 100 simulations with a random initial state. Table 2 indicates the win, lose, and draw times and the average time consumed by each decision-making of the proposed method. As shown in Table 2, on the one hand, the proposed method won five more times than the MCTS method, and 60 simulations is drawn. These results indicate that the proposed method is feasible and effective, even though the proposed method is just slightly better. On the other hand, when simulations started from a random initial state, the proposed method is almost the same as the MCTS method, which indicates that the initial state has a significant influence on the decision-making method. As can be seen from Table 2, for the proposed method, the average time taken for each decision-making is ∼0.38 s. We also compute the average time of the original MCTS method (He et al., 2017), which is 0.11 s, this means that the proposed method is slower.

Next, we show the process of the MCTS maneuver decision-making method in continuous action space, then we show the process of the method we proposed by a fight against the MCTS method. The initial position of aircraft 1 is (40,000, 40,000, 10,000), the pitch angle is 0°, the yaw angle is 180°, and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10,000), the pitch angle is 0°, the yaw angle is 0°, and the initial velocity is 300 m/s. Aircraft 1 moves at a constant speed in a straight line, and aircraft 2 maneuvers using the MCTS method with human knowledge. The simulation result is shown in Figure 7.

Figure 7A shows the trajectory of both sides, in which the blue solid line represents the flight trajectory of aircraft 1 and the orange solid line represents the flight trajectory of aircraft 2. In Figure 7B, the solid blue line indicates the velocity change of aircraft 1 and the orange solid line represents the velocity change of aircraft 2. Figure 7C indicates the overload change of missiles of aircraft 2. It can be seen that the MCTS method with human knowledge can react to the aircraft with simple maneuver and at the end of the simulation, the missile of aircraft 2 hit the target, which suggests the effectiveness of the MCTS method.

In simulation 2, aircraft 1 uses the proposed method and aircraft 2 uses the MCTS method (He et al., 2017), but the action space of the two methods is the same. As described in He et al. (2017), it combines the angle advantage function, distance advantage function, velocity advantage function, and height advantage function with MCTS, which means that it makes maneuver decisions with human knowledge. These advantage functions which stem from human knowledge can guide the aircraft to approach the target. However, our method uses only the final result r_{T} = {-1, 0, 1}, as described in Section Reinforcement learning from self-play, including no human knowledge.

The initial position of aircraft 1 is (70,000, 70,000, 10,000), the pitch angle is 0°, the yaw angle is 180°, and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10,000), the pitch angle is 0°, the yaw angle is 0°, and the initial velocity is 300 m/s. As can be seen that the initial situation of both sides is equal. The simulation result is shown in Figure 8.

Figure 8A shows the trajectory of both sides, in which the blue solid line represents the flight trajectory of aircraft 1, the orange solid line represents the flight trajectory of aircraft 2, the green dotted line represents the flight trajectory of missile 1, and the red dotted line represents the flight trajectory of missile 2. In Figure 8B, the solid blue line indicates the velocity change of aircraft 1 and the orange solid line represents the velocity change of aircraft 2. Figure 8C indicates the overload change of missiles of the two sides, and it can be seen from Figure 8C that the missile overload is small when it is far from the target and reaches the maximum when it hit the target.

As can be seen from Figure 8A, when the simulation begins, both sides deflect toward each other and launch missiles, but their decision-making principles are different: aircraft 1 concludes that deflecting to aircraft 2 is of high value according to a large number of self-play data, while aircraft 2 chooses to deflect to aircraft 1 because it can increase the value of the air combat advantage function. In the end, the missile of aircraft 1 hit aircraft 2, and the distance between missile 2 and aircraft 1 is about 8 km. This suggests that the proposed method without human knowledge is stronger.

The initial position of aircraft 1 is (80 000, 80 000, 10 000), the pitch angle is 0°, the yaw angle is 180°, and the initial velocity is 300 m/s. The initial position of aircraft 2 is (0, 0, 10 000), the pitch angle is 0°, the yaw angle is 45°, and the initial velocity is 300 m/s. As can be seen that the initial situation of aircraft 2 is at an advantage. The simulation result is shown in Figure 9.

The simulation ended because the altitude of aircraft 2 exceeded the maximum altitude. The air combat advantage function of aircraft 2 includes the constraint on flight altitude to keep the altitude difference between it and the target within a certain range. However, although it used the advantage function based on human knowledge to guide maneuver decision-making, it failed to control the flight altitude properly because of the randomness of MCTS. On the contrary, the proposed method also based on MCTS can keep the flight altitude within a reasonable range without human knowledge, which indicates the effectiveness of the proposed method.

At the same time, according to Figures 7B, 8B, 9B, it can be seen that decision-making guided by human knowledge always increases the speed, while decision-making without human knowledge accelerates and decelerates, which shows that the method without human knowledge is more reasonable. Because the maximum speed is set as 420 m/s, it can be seen from the speed-increasing trend in Figures 7B, 8B, 9B that if the maximum speed is not set, the decision-making method guided by human knowledge will continue to increase the speed and always maintain the maximum speed in the subsequent air combat, which is not reasonable. Therefore, the method proposed in this paper without human knowledge is more reasonable and effective.

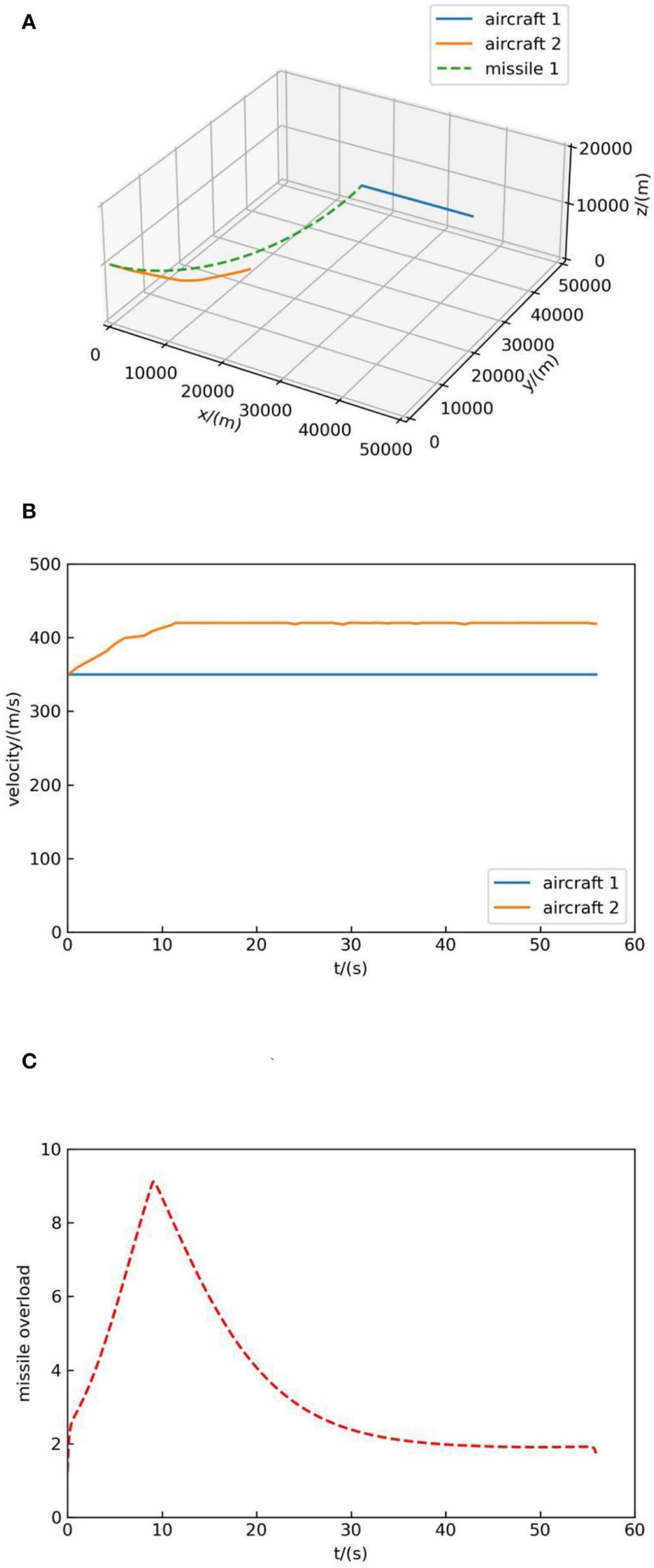


FIGURE 7
Simulation result 1. **(A)** Air combat trajectory. **(B)** Velocity. **(C)** Missile overload.

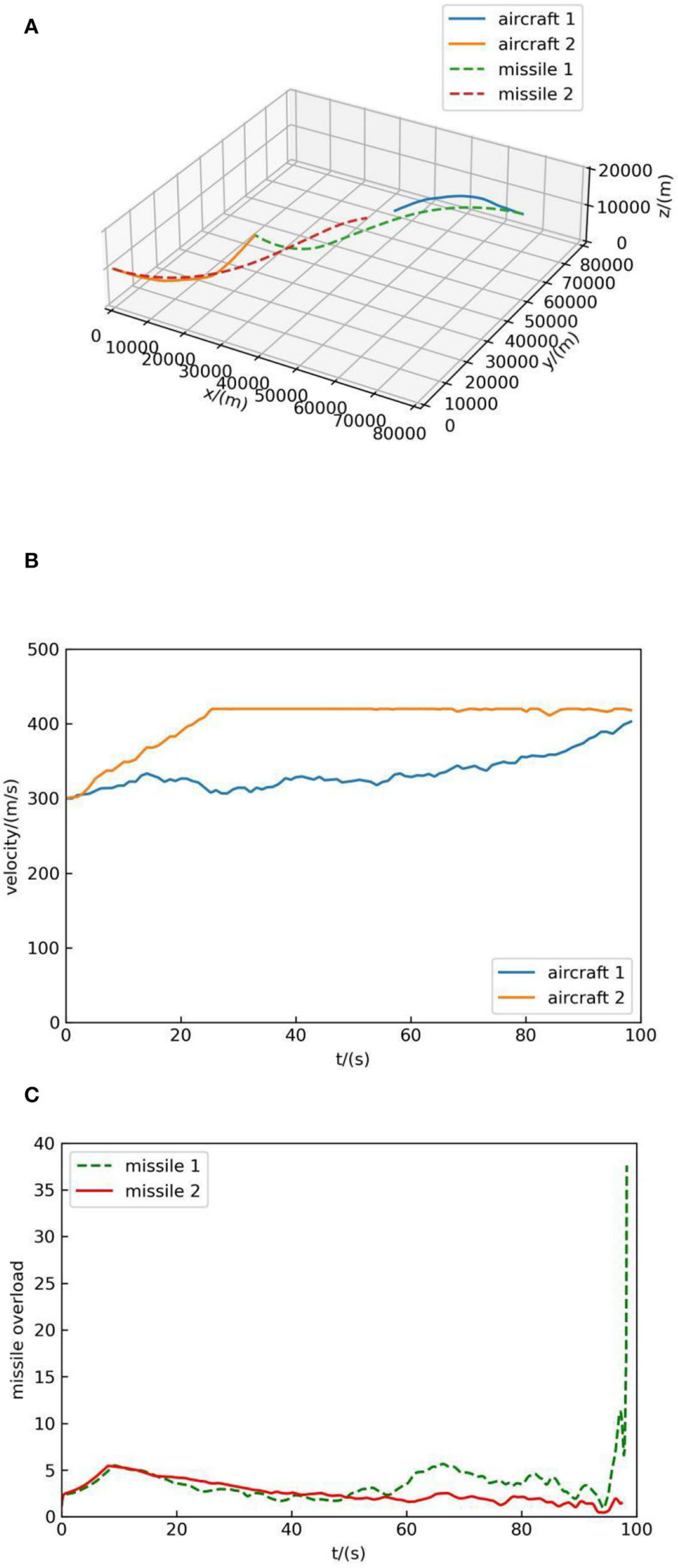


FIGURE 8
Simulation result 2. **(A)** Air combat trajectory. **(B)** Velocity. **(C)** Missile overload.

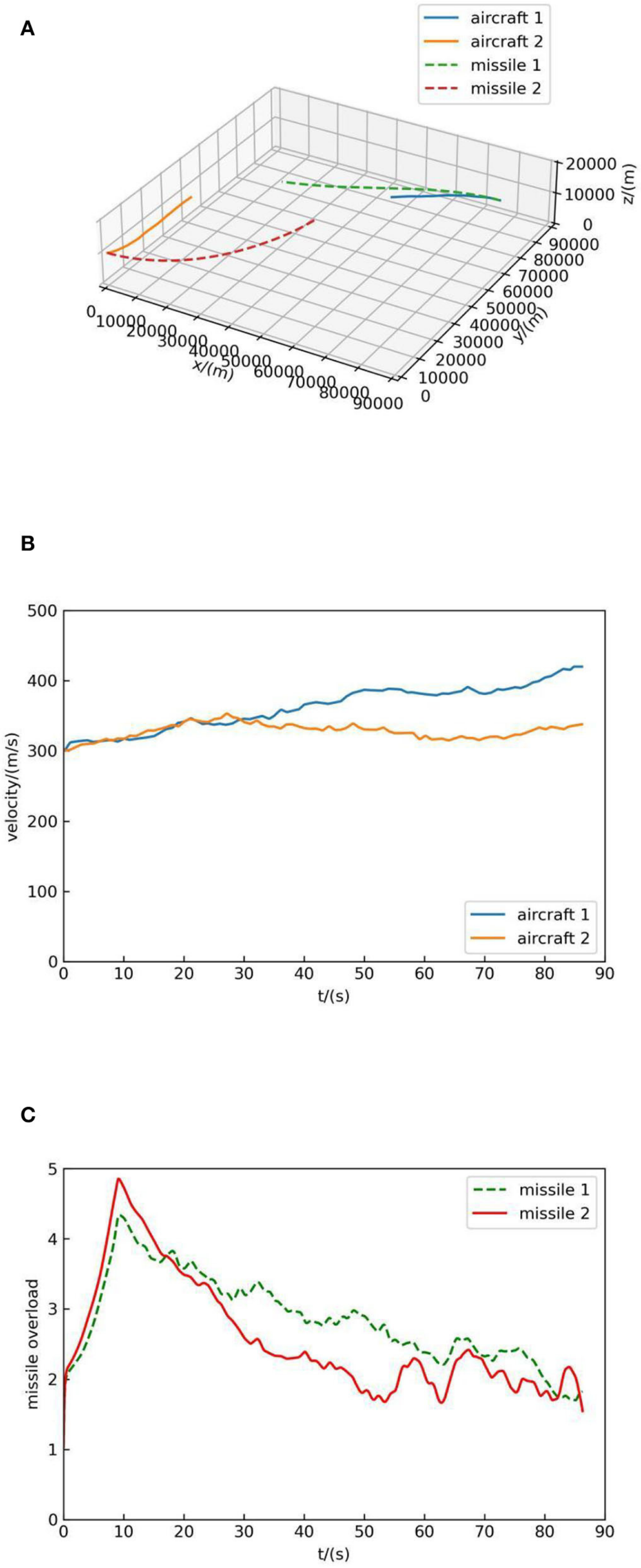


FIGURE 9
Simulation result 3. **(A)** Air combat trajectory. **(B)** Velocity. **(C)** Missile overload.

Conclusion

The maneuver decision-making method based on deep reinforcement learning and Monte Carlo tree search without human knowledge is proposed in this paper. According to the simulation results, it can be concluded that a pure reinforcement learning approach without human knowledge is feasible and efficient for autonomous air combat maneuver decision-making. On the one hand, the strengths of the proposed method are as follows: (1) The method can achieve similar performance as the method with human knowledge. (2) The method is simple to implement since elaborately designed reward based on human knowledge is not necessary. (3) The method can train neural networks from scratch without using any data from human pilots, which indicates that it can be used in the domains where human data are deficient or expensive to acquire. On the other hand, the weaknesses of the proposed method are as follows: (1) The performance of the method is not as good as its counterparts in board games, such as Go and chess. (2) The time consumption of the method is more than some traditional methods. (3) It takes plenty of time for training an agent using this method.

We aim to investigate whether it is feasible for maneuver decision-making without human knowledge by means of simulations and using the results for a recommendation system or pilots in manned aircraft is out of the scope of the article. In future work, considering AlphaGo Zero without human knowledge can defeat previous algorithms and human players in Go, and it is necessary to improve the performance of the method without human knowledge since the proposed method does not completely defeat the methods with human knowledge. Meanwhile, decreasing the time consumption of the method is also another future work because the time consumption of the proposed method is more than some traditional methods. And the training procedure needs to be improved since it takes plenty of time for training an agent.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62101590) and the Natural Science Foundation of Shaanxi Province (Grant No. 2020JQ-481).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Baxter, J., Tridgell, A., and Weaver, L. (2000). Learning to play chess using temporal differences. *Mach. Learn.* 40, 243–263. doi: 10.1023/A:1007634325138

Du, H. W., Cui, M. L., Han, T., Wei, Z., Tang, C., and Tian, Y. (2018). Maneuvering decision in air combat based on multi-objective optimization and reinforcement learning. *J. Beij. Uni. Aero. Astronau.* 44, 2247–2256. doi: 10.13700/j.bh.1001-5965.2018.0132

Eloy, G., David, W. C., Dzung, T., and Meir, P. (2020). "A differential game approach for beyond visual range tactics," in *2021 American Control Conference* (New Orleans, LA).

Fang, X., Liu, J., and Zhou, D. (2019). Background interpolation for on-line situation of capture zone of air-to-air missiles. *J. Syst. Eng. Electron.* 41, 1286–1293. doi: 10.3969/j.issn.1001-506X.2019.06.16

Goodfellow, I., Bengio, Y., and Courville, A. (2017). *Deep Learning.* Beijing: Posts Telecom Press.

Guo, H., Hou, M., Zhang, Q., and Tang, C. (2017). UCAV robust maneuver decision based on statistics principle. *Acta Arma.* 38, 160–167. doi: 10.3969/j.issn.1000-1093.2017.01.021

Hado, H., Arthur, G., and David, S. (2016). "Deep reinforcement learning with double q-learning," in *National Conference of the American Association for Artificial Intelligence* (Phoenix, AZ: PMLR), 1813–1825.

He, X., Jing, X., and Feng, C. (2017). Air combat maneuver decision based on MCTS method. *J. Air For. Eng. Uni.* 18, 36–41. doi: 10.3969/j.issn.1002-0640.2018.03.008

Hinton, G., and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

Hu, D., Yang, R., Zuo, J., Zhang, Z., Wu, J., and Wang, Y. (2021). Application of deep reinforcement learning in maneuver planning of beyond-visual-range air combat. *IEEE Access.* 9, 32282–32297. doi: 10.1109/ACCESS.2021.3060426

Huang, C., Dong, K., Huang, H., Tang, S., and Zhang, Z. (2018). Autonomous air combat maneuver decision using Bayesian inference and moving horizon optimization. *J. Syst. Eng. Electron*. 29, 86–97. doi: 10.21629/JSEE.2018.01.09

Jonas, D., Federico, F., and Jonas, B. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* 602, 414–419. doi: 10.1038/s41586-021-04301-9

Li, S., Ding, Y., and Gao, Z. (2019). UAV air combat maneuvering decision based on intuitionistic fuzzy game theory. *J. Syst. Eng. Electron*. 41, 1063–1070. doi: 10.3969/j.issn.1001-506X.2019.05.19

Li, W., Shi, J., and Wu, Y. (2020). A multi-UCAV cooperative occupation method based on weapon engagement zones for beyond-visual-range air combat. *Def. Tech*. 4, 1–17. doi: 10.1016/j.dt.2021.04.009

Li, X., Zhou, D., and Feng, Q. (2015). Air-to-air missile launch envelops fitting based on genetic programming. *J. Project. Rockets Missile Guid*. 35, 16–18. doi: 10.15892/j.cnki.djzdxb.2015.03.005

Li, Z. (2010). China radar guided air-to-air missile. *Shipborne Weap*. 2, 22–35. doi: 10.15892/j.cnki.jzwq.2010.02.013

Ma, X., Li, X., and Zhao, Q. (2018). "Air combat strategy using deep Q-learning," in *Chinese Automation Congress*, 3952–3957. doi: 10.1109/CAC.2018.8623434

Ma, Y., Wang, G., Hu, X., Luo, H., and Lei, X. (2020). Cooperative occupancy decision making of multi-UAV in beyond-visual-range air combat: a game theory approach. *IEEE Access*. 8, 11624–11634. doi: 10.1109/ACCESS.2019.2933022

Matteo, H., Joseph, M., and Hado, H. (2017). *Rainbow: Combining Improvements in Deep Reinforcement Learning*. Available online at: http://arxiv.org/abs/1710.02298v1.

Mcgrew, J. S., How, J. P., and Williams, B. (2010). Air-combat strategy using approximate dynamic programming. *J. Guid. Control Dynam*. 33, 1641–1654. doi: 10.2514/1.46815

Mnih, V., Kavukcuoglu, K., and Silver, D. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Moravcík, M. (2017). DeepStack: expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 508–513. doi: 10.1126/science.aam6960

Oriol, V., Igor, B., and Silver, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 350–354. doi: 10.1038/s41586-019-1724-z

Rosin, D. (2011). Multi-armed bandits with episode context. *Ann. Math. Artif. Intell*. 61, 203–230. doi: 10.1007/s10472-011-9258-6

Schrittwieser, J., Antonoglou, I., and Silver, D. (2020). Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature* 588, 604–609. doi: 10.1038/s41586-020-03051-4

Sheppard, B. (2002). World-championship-caliber Scrabble. *Artif. Intell*. 134, 241–275. doi: 10.1016/S0004-3702(01)00166-7

Silver, D., Huang, A., and Maddison, C. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961

Silver, D., Schrittwieser, J., and Simonyan, K. (2017). Mastering the game of Go without human knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270

Wang, J., Ding, D., Xu, M., Han, B., and Lei, L. (2019). Air-to-air missile launchable area based on target escape maneuver estimation. *J. Beij Uni. Aero Astronau*. 45, 722–734. doi: 10.13700/j.bh.1001-5965.2018.0462

Watkins, H., and Dayan, P. (1992). Q-learning, *Mach. Learn*. 8, 279–292. doi: 10.1007/BF00992698

Wei, R. X., Zhou, K., Ru, C. J., Guan, X., and Che, J. (2015). Study on fuzzy cognitive decision-making method for multiple UAVs cooperative search. *Sci. Sin. Tech*. 45, 595–601. doi: 10.1360/N092015-00130

Williams, P. (1990). Three-dimensional aircraft terrain-following via real-time optimal control. *J. Guid. Control Dynam*. 13, 1146–1149.

Wu, S., and Nan, Y. (2013). The calculation of dynamical allowable lunch envelope of air-to-air missile after being launched. *J. Project. Rockets Missile Guid*. 33, 49–54. doi: 10.15892/j.cnki.djzdxb.2013.05.012

Wu, W., Zhou, S., Gao, L., and Liu, J. (2011). Improvements of situation assessment for beyond-visual-range air combat based on missile launching envelope analysis. *J. Syst. Eng. Electron*. 33, 2679–2685. doi: 10.3969/j.issn.1001-506X.2011.12.20

Yang, Z., Zhou, D., Piao, H., Zhang, K., Kong, W., and Pan, Q. (2020). Evasive maneuver strategy for UCAV in beyond-visual-range air combat based on hierarchical multi-objective evolutionary algorithm. *IEEE Access*. 8, 46605–46623. doi: 10.1109/ACCESS.2020.2978883

You, Y., Li, J., and Shen, L. (2019). An effective Bayesian network parameters learning algorithm for autonomous mission decision-making under scarce data. *Int. J. Mach. Learn. Cyber*. 10, 549–561. doi: 10.1007/s13042-017-0737-x

Zhang, P., Song, C., and Zhang, J. (2015). Development analysis of radar air-to-air missile. *Aerodyn. Missile J*. 4, 30–33. doi: 10.15892/j.cnki.fhdd.2015.04.008

Zhang, Q., Yang, R., Yu, L., Zhang, T., and Zuo, J. (2018). BVR air combat maneuvering decision by using Q-network reinforcement learning. *J. Air For. Eng. Uni*. 19, 8–14. doi: 10.3969/j.issn.1009-3516.2018.06.002

Zhou, T., Chen, M., Wang, Y., He, J., and Yang, C. (2020a). Information entropy-based intention prediction of aerial targets under uncertain and incomplete information. *Entropy* 22, 1–19. doi: 10.3390/e22030279

Zhou, T., Chen, M., Yang, C., and Nie, Z. (2020b). Data fusion using Bayesian theory and reinforcement learning method. *Sci. China Inform. Sci*. 63, 170209. doi: 10.1007/s11432-019-2751-4

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 13 December 2022
DOI 10.3389/fnbot.2022.1012427

Check for updates

OPEN ACCESS

EDITED BY
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

REVIEWED BY
Zhijun Yang,
Middlesex University, United Kingdom
Önder Tutsoy,
Adana Science and Technology University, Turkey

*CORRESPONDENCE
Guilin Wen
glwen@hnu.edu.cn;
glwen@ysu.edu.cn

RECEIVED 05 August 2022
ACCEPTED 23 November 2022
PUBLISHED 13 December 2022

CITATION
Pan Z, Wen G, Tan Z, Yin S and Hu X (2022) An immediate-return reinforcement learning for the atypical Markov decision processes. *Front. Neurorobot.* 16:1012427. doi: 10.3389/fnbot.2022.1012427

An immediate-return reinforcement learning for the atypical Markov decision processes

Zebang Pan[1], Guilin Wen[1,2]*, Zhao Tan[1], Shan Yin[1,2] and Xiaoyan Hu[1]

[1]State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha, Hunan, China, [2]School of Mechanical Engineering, Yanshan University, Qinhuangdao, Hebei, China

The atypical Markov decision processes (MDPs) are decision-making for maximizing the immediate returns in only one state transition. Many complex dynamic problems can be regarded as the atypical MDPs, e.g., football trajectory control, approximations of the compound Poincaré maps, and parameter identification. However, existing deep reinforcement learning (RL) algorithms are designed to maximize long-term returns, causing a waste of computing resources when applied in the atypical MDPs. These existing algorithms are also limited by the estimation error of the value function, leading to a poor policy. To solve such limitations, this paper proposes an immediate-return algorithm for the atypical MDPs with continuous action space by designing an unbiased and low variance target Q-value and a simplified network framework. Then, two examples of atypical MDPs considering the uncertainty are presented to illustrate the performance of the proposed algorithm, i.e., passing the football to a moving player and chipping the football over the human wall. Compared with the existing deep RL algorithms, such as deep deterministic policy gradient and proximal policy optimization, the proposed algorithm shows significant advantages in learning efficiency, the effective rate of control, and computing resource usage.

KEYWORDS

reinforcement learning, atypical Markov decision process, flight trajectory control, uncertain environments, continuous action space

Introduction

Inspired by the learning pattern of humans, i.e., learning by interacting with the external environment, the concepts of reinforcement learning (RL) were first proposed by Minsky (1954). Subsequently, Bellman (1957) presented a method to define an RL problem using Markov decision processes (MDPs). As a result, an RL problem can be described clearly in terms of states, actions, and rewards. In recent years, with an in-depth combination of deep learning, traditional RL has evolved into deep RL. Generally speaking, deep RL algorithms can be subdivided into value-based algorithms and policy gradient algorithms. Deep Q Network (DQN) was the first exploration for value-based algorithms (Mnih et al., 2015). It solved the dimension explosion problem. Subsequently, various improved DQN algorithms were developed, such as Double DQN (Van Hasselt et al., 2016), Dueling DQN (Wang et al., 2016), etc. However, value-based algorithms could only be applied in discrete rather than continuous action space. In contrast, policy gradient algorithms could solve the RL problem with continuous action space, as an independent actor was constructed to output actions. Note that policy gradient algorithms were generally divided into stochastic policy algorithms and deterministic policy algorithms. The stochastic policy algorithms could output the probability distribution of the actions, such as the asynchronous advantage actor-critic (A3C) (Mnih et al., 2016) and proximal policy optimization (PPO) (Schulman et al., 2017). The deterministic policy algorithms could output the deterministic actions, such as deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015). Due to the advantages of model-free, great self-learning ability, etc., the RL has shown excellent performance in the application of complex control processes. For example, the RL methods were applied to robot manipulators to solve trajectory planning under complex environments (Chen et al., 2022). Tutsoy and Brown studied the RL in problems with Chaotic dynamics and proved that a reasonable discount factor could avoid singular learning problems (Tutsoy and Brown, 2016). Pan et al. (2023) designed a controller for a three-link biped robot using the twin delayed deep deterministic policy gradient algorithm (TD3). Sharbafi et al. designed controllers based on the RL for their football robots and won third place in the 2011 world games (Sharbafi et al., 2011). Massi et al. (2022) increase the learning speed of a navigating robot to improve its performance using the RL method. Even in the financial sector, the RL could be used to learn investment trading policy (Lee et al., 2021). Such trading systems based on RL improved trading performance effectively.

Indeed, the above application scenarios belong to the standard MDPs, containing a series of state transitions. However, the atypical MDP case, which involves only one state transition in continuous action space, can also arise in the engineering field, such as the stamping process (Wang and Budiansky, 1978), directional blasting (Zhu et al., 2008), football trajectory control (Myers and Mitchell, 2013), approximations of the compound Poincaré maps (Li et al., 2020), etc. In such atypical MDPs, the control goal is to maximize the immediate returns rather than the long-term returns. Therefore, compared to the standard MDPs, the atypical MDPs can exhibit many new characteristics. Furthermore, to the best knowledge of the authors, all existing RL algorithms are designed for the standard MDPs to maximize long-term returns. Applying the existing RL algorithms to the atypical MDPs shall lead to the following problems. On the one hand, the existing RL algorithms are also limited by their open problem, i.e., the estimation error of the value function. For example, the sampling errors caused by incomplete samplings will lead to bias for the estimated state-value function (e.g., A3C and PPO) (Mnih et al., 2016; Schulman et al., 2017). For the estimated action-value function, DQN and DDPG can cause the overestimation due to the max operation in off-policy temporal-difference (TD) learning (Mnih et al., 2015; Van Hasselt et al., 2016). In comparison, the TD3 and double DQN may lead to underestimation as the minimum output of two independent target critic networks is selected to update the action-value function (Lillicrap et al., 2015; Fujimoto et al., 2018). Furthermore, the uncertain environment may bring a high variance for the estimated value functions as the uncertainties can lead to entirely different rewards for the same state-action pair. Since the policy gradient formulation is directly related to the value function, the estimation error of the value function can lead to a poor policy and limit the performance of the existing RL algorithms. On the other hand, as the atypical MDPs focus only on immediate returns, the common designs for calculating long-term returns are redundant in the existing RL algorithms. It may result in a waste of computing resources. Moreover, existing algorithms do not notice the difference between estimating the state-value function and the action-value function in atypical MDPs. Such difference determines which approach is more suitable for dealing with atypical MDPs. Thus, regarding the above problems of the existing RL algorithms, this paper aims to propose an immediate-return RL algorithm for atypical MDPs with continuous action space.

On this basis, this paper further takes the football trajectory control as the illustration example to present the superior performance of the proposed algorithm. Indeed, the football trajectory control shall be an ideal test case for the proposed algorithm. The reasons are as follows. As the whole process contains only one state transition from take-off to end and its action, i.e., the football's initial velocity, is continuous, football flight is an atypical MDP case with continuous action space. Meanwhile, the aerodynamic model of football is strongly non-linear and has no analytical solutions (Myers and Mitchell, 2013; Javorova and Ivanov, 2018), which involves many complex physical laws (Horowitz and Williamson, 2010; Norman and McKeon, 2011; Javorova and Ivanov, 2018; Kiratidis and Leinweber, 2018). It is difficult for the traditional control method to control football flight (Hou and Wang, 2013; Hou et al., 2016). Thus, as a challenging task, football trajectory control is an ideal example to test the proposed algorithm. In addition, related researches also have practical application value. The accuracy of the shot is a key of the football robot. Designing a high-performance controller based on the proposed algorithm can promote the development of high-level football robots in the Robot world cup (Sharbafi et al., 2011).

The main contents and contributions of this paper are summarized as the following aspects. Firstly, the characteristics of the atypical MDPs are analyzed systematically based on the RL theory. The disadvantage of estimating the state-value function in the atypical MDPs is explained qualitatively, i.e., the large samples requirement and the unavoidable sampling error. These studies indicate the way to the development of RL algorithms in the atypical MDPs. That is, the deterministic policy has natural advantages in dealing with the atypical MDPs in continuous action space. Secondly, based on the deterministic policy and estimated action-value function, an immediate-return RL algorithm is proposed for the atypical MDPs. In the proposed algorithm, the average reward method is developed to construct an unbiased and low variance target Q-value. Compared with existing RL algorithms, e.g., DDPG and PPO, the proposed algorithm reduces the estimation error significantly. More details are introduced in following Section Immediate-return RL algorithm for the atypical MDPs. Meanwhile, a simplified network framework is also designed for the proposed algorithm. Thus, the proposed decreases both the space complexity and time complexity. The comparison tests also demonstrate that the computing resource consumed by the proposed algorithm is lower than the DDPG and PPO. Thirdly, two challenging scenarios of the football trajectory control, i.e., passing the football to a moving player, and chipping the football over the human wall (chip kick), are presented to test the feasibility of the proposed algorithm. These scenarios can be used as the benchmark to test the algorithms designed for the atypical MDPs. Meanwhile, the controllers based on the proposed algorithm in this paper can improve the football robot's shot accuracy in competitions, such as the Robot world cup (Sharbafi et al., 2011). In the above scenarios, existing RL algorithms (i.e., DDPG, PPO) are also tested as references. Numerical results demonstrate that the immediate-return RL algorithm has higher learning efficiency, a higher effective rate of control, and lower computing resource usage than the reference RL algorithms.

The rest of the present work is organized as follows. In Section The atypical MDPs, the analysis of the atypical MDPs is introduced. Then, the immediate-return RL algorithm for the atypical MDPs is proposed in Section Immediate-return RL algorithm for the atypical MDPs. In Section Illustration examples: Football trajectory control for different scenarios, two illustration examples in MDPs, i.e., passing the football to a moving player and chipping the football over the human wall, are designed. In Section Comparison and discussion, the feasibility and high performance of the RL controllers are demonstrated by simulation tests. And the advantages of the immediate-return RL algorithm are discussed by comparison with the existing RL algorithms. Lastly, the conclusion of this paper is drawn in Section Conclusion.

The atypical MDPs

Atypical MDPs: Definition and characteristic analyses

For the standard MDP, it can be described by the states s_t, actions a_t, and rewards r_t (immediate return). Thus, the trajectory of a standard MDP case contains a series of contiguous state transitions, which can be expressed as follows.

$$(s_0,\ a_0,\ r_0) \rightarrow \ \ldots \rightarrow (s_t,\ a_t,\ r_t) \rightarrow (s_{t+1},\ a_{t+1},\ r_{t+1}) \rightarrow \ldots \rightarrow s_{ter} \tag{1}$$

where s_{ter} is the termination state. Based on RL theory, the state-value function V_π and action-value function Q_π in standard MDPs is defined as follows (Watkins, 1989; Sutton and Barto, 2018).

$$V_\pi\ (s_t)\ =\ \sum\nolimits_{a_t} \pi\left(a_t|s_t\right) \sum\nolimits_{s_{t+1},r_t} p\left(s_{t+1}, r_t \middle| s_t, a_t\right) \left[r_t + \gamma V_\pi\ \left(s_{t+1}\right)\right] \tag{2}$$

$$Q_\pi\ (s_t,\ a_t)\ =\ \sum\nolimits_{s_{t+1},r_t} p\left(s_{t+1}, r_t \middle| s_t, a_t\right) \left[r_t + \gamma \sum\nolimits_{a_{t+1}} \pi\left(a_{t+1}|s_{t+1}\right) Q_\pi\ \left(s_{t+1}, a_{t+1}\right)\right] \tag{3}$$

where p is the state transition probability and γ is the reward discount factor (Sutton and Barto, 2018). As shown in Equations (2), (3), both $V_\pi\ (s_t)$ and $Q_\pi\ (s_t,\ a_t)$ are closely related to the value of its possible successor states (or state-action pairs) (Sutton and Barto, 2018). Then, the control goal in a standard MDP case is achieving the optimal expected long-term returns. The optimal policy π^* can be written as follows (Sutton and Barto, 2018).

$$\pi^*\ (s_t) = \mathrm{argmax}_{a_t \epsilon A} Q_{\pi^*}\ (s_t,\ a_t) \tag{4}$$

In contrast, the atypical MDP case considered in this paper involves continuous action space and has only one state transition from the initial state s_t $(t = 0)$ to the termination state s_{ter}. That is, for any state s_t, its next state s_{t+1} is identical to the termination state s_{ter} after a state transition, i.e., $s_{t+1} \equiv s_{ter}$. Its trajectory can be expressed as follows.

$$(s_t,\ a_t,\ r_t) \rightarrow s_{ter} \tag{5}$$

As defined in Equation (5), due to $s_{t+1} \equiv s_{ter}$, the whole process of an atypical MDP case only contains one reward r_t (immediate return). Thus, in the atypical MDPs, only the immediate return rather than the long-term return should be considered. Note that the atypical MDP case involving continuous action space is common in engineering field, e.g., stamping process, directional blasting, football trajectory control, approximations of the compound Poincaré maps, etc.

Then, the characteristics of atypical MDPs will be analyzed by comparing the differences between the standard value functions in Equations (2), (3) and the value functions of the atypical MDPs. As defined by Sutton et al., both the state-value and the Q-value at the termination state s_{ter} are identical to zero (Sutton and Barto, 2018), i.e., $V_\pi\ (s_{ter}) \equiv 0$ and $Q_\pi\ (s_{ter}) \equiv 0$. Since $s_{t+1} \equiv s_{ter}$ in the atypical MDPs, the state-value function V_π^A in the atypical MDPs can be written as follows.

$$\begin{aligned} V_\pi^A\,(s_t) &= \sum\nolimits_{a_t} \pi\,(a_t|s_t) \sum\nolimits_{s_{t+1},r_t} p\left(s_{t+1}, r_t \middle| s_t, a_t\right) r_t \\ &= \sum\nolimits_{a_t} \pi\,(a_t|s_t) R(s_t, a_t) \end{aligned} \tag{6}$$

In atypical MDPs, $V_\pi^A\,(s_t)$ denotes the expected immediate return of the state s_t under the policy π. $R(s_t, a_t)$ is the expected immediate return for the state-action pairs. Compared to the V_π in standard MDPs [see Equation (2)], although computing the value of V_π^A in the atypical MDPs is independent of its successor state-value $V_\pi\,\left(s_{t+1}\right)$, V_π^A is still a function of the policy π in the atypical MDPs. Due to the operation $\sum_{a_t} \pi(a_t|s_t)$ in Equation (6), estimating $V_\pi^A\,(s_t)$ should traverse the whole action space A under the current policy π. It means that approximating the $V_\pi^A\,(s_t)$ requires large amounts of samplings when the policy π is stochastic. A finite number of samplings may ignore the huge un-sampled action space and cause an enormous sampling error. Here, suppose that the whole action space A consists of the sampled action space A^s and the un-sampled action space A^{un}, i.e., $A = A^s + A^{un}$. Based on Equation (6), there must be a sampling error $err(s_t)$ between the estimated state-value function V_π^E and true state-value function V_π^A, i.e.,

$$V_\pi^A\,(s_t) = V_\pi^E\,(s_t) + err(s_t) \tag{7}$$

where, $V_\pi^E\,(s_t)$ and $err(s_t)$ can be expressed as follows:

$$V_\pi^E\,(s_t) = \sum\nolimits_{a_t \in A^s} \pi(a_t|s_t) R(s_t, a_t) \tag{8}$$

$$err(s_t) = \sum\nolimits_{a_t \in A^{un}} \pi(a_t|s_t) R(s_t, a_t) \tag{9}$$

Actually, in standard MDPs, such sampling errors also exist in the estimation of the V_π and Q_π since they are also the functions of the policy π. This sampling error introduces the bias for the estimated $V_\pi^E\,(s_t)$ and further negatively affect the stochastic policy update. Based on the actor-critic method with baseline (Sutton and Barto, 2018; Levine et al., 2020), the estimated stochastic policy gradient $\hat{g}^E$ can be written as follows when the biased estimate V_π^E is used (Sutton et al., 1999; Schulman, 2016).

$$\begin{aligned} \hat{g}^E &= E\left[\sum_{t=0}^{\infty} (r_t + \gamma V_\pi^E\,(s_{t+1}) - V_\pi^E\,(s_t)) \nabla_\omega \log \pi_\omega(a_t|\,s_t)\right] \\ &= E\Bigg[\sum_{t=0}^{\infty} \left(r_t + \gamma\left(V_\pi^A\,(s_{t+1}) - err\,(s_{t+1})\right)\right) - \left(V_\pi^A\,(s_t)\right. \\ &\quad \left. -err(s_t)\right)) \nabla_\omega \log \pi_\omega(a_t|\,s_t)\Bigg] \\ &= E\Bigg[\sum_{t=0}^{\infty} \left((r_t + \gamma V_\pi^A\,(s_{t+1}) - V_\pi^A\,(s_t)) - (\gamma err\,(s_{t+1})\right. \\ &\quad \left. -err\,(s_t))\right) \nabla_\omega \log \pi_\omega(a_t|\,s_t)\Bigg] \\ &= \hat{g} + E\left[\sum_{t=0}^{\infty} (err\,(s_t) - \gamma err(s_{t+1})) \nabla_\omega \log \pi_\omega(a_t|\,s_t)\right] \end{aligned} \tag{10}$$

where $\hat{g}$ is the true stochastic policy gradient. The biased estimate V_π^E causes an ineradicable policy gradient error $\hat{g}_{err}$ between the estimated $\hat{g}^E$ and true $\hat{g}$, i.e.,

$$\hat{g}_{err} = E\left[\sum_{t=0}^{\infty} (err\,(s_t) - \gamma err(s_{t+1})) \nabla_\omega \log \pi_\omega(a_t|\,s_t)\right] \tag{11}$$

This error $\hat{g}_{err}$ may cause negative effects on policy updates.

Under the theory of RL, the action-value function Q^A in the atypical MDPs can be written as follows.

$$Q^A\,(s_t,\ a_t) = \sum\nolimits_{s_{t+1},r_t} P\left(s_{t+1}, r_t \middle| s_t, a_t\right) r_t = R(s_t, a_t) \tag{12}$$

In the atypical MDPs, $Q^A\,(s_t,\ a_t)$ denotes the expected immediate return of the state-action pairs $(s_t,\ a_t)$. And the action-value function Q^A is also unrelated to the value of its successor state-action pairs as same as the V_π^A in Equation (6). However, it should be particularly stressed that the action-value function Q^A in the atypical MDPs is a function independent of policy π, which is different from the V_π in Equation (2), Q_π in Equation (3), and V_π^A in Equation (6). Thus, it brings a set of new characteristics for the Q^A as follows. Firstly, the value of the $Q^A\,(s_t,\ a_t)$ will not be changed in policy updating. However, with the policy π updating, the state-value function V_π^A in the atypical MDPs will be changed accordingly. That is, compared to approximating the Q^A, approximating the V_π^A requires more samples and more training steps. Meanwhile, since there is no $\sum_{a_t} \pi\,(a_t|s_t)$ operation in Equation (12), it is unnecessary for estimating the action-value function Q^A to traverse the whole action space. It also indicates that much more samples are required to estimate $V_\pi^A\,(s_t)$ than to estimate $Q^A\,(s_t,\ a_t)$ in an atypical MDP case. This also indicates that estimating the $V_\pi^A\,(s_t)$ in an atypical MDPs requires more samples than the Q-function. Thus, estimating the state-value function can lead to the low learning efficiency of the RL algorithms. Secondly, the bias caused by sampling error will not exist in the estimated action-value function Q^E as Equation (12) does not contain operation $\sum_{a_t} \pi\,(a_t|s_t)$. In contrast, such bias is inevitable for the estimated state-value function V_{π}^{E}, as discussed in Equation (7). Based on the above analysis, estimating the $Q^{A}\left(s_t,\ a_t\right)$ is easier than estimating the $V_{\pi}^{A}\left(s_t\right)$ in the atypical MDPs. Generally speaking, the stochastic policy algorithms rely on the estimated state-value function V_{π}^{E}, and the deterministic policy algorithms rely on the estimated action-value function Q^{E}. Thus, when dealing with the atypical MPD case, deterministic policy algorithms can show more natural advantages than the stochastic policy algorithms.

In addition, the new characteristic of the atypical MDP case is also shown in its policy π^{*}. Based on the definition of the action-value function Q^{A} in Equation (12), the optimal policy π^{*} under the atypical MDPs can be expressed as follows.

$$\begin{aligned}\pi^{*}\left(s_t\right) &= \underset{a_t\in A}{\operatorname{argmax}}\sum\nolimits_{s_{t+1},r_t} P\left(s_{t+1},r_t\middle|s_t,a_t\right)r_t \\ &= \underset{a_t\in A}{\operatorname{argmax}}R\left(s_t,\ a_t\right)\end{aligned} \tag{13}$$

That is, in the atypical MDPs, the goal of the optimal policy π^{*} is achieving the maximal expected reward rather than the maximal expected long-term returns. And the long-term returns can be ignored for policy update in the atypical MDPs.

Limitations of existing RL algorithms in the atypical MDPs

When dealing with the atypical MDP cases in continuous action space, the existing RL algorithms are limited by their open problems as well as by the special problems caused by the characteristic of the atypical MDP case. Note that the value-based algorithms will not be discussed here as they are only applicable to discrete action space.

The estimation error of the estimated value function, i.e., bias and variance, is an open problem that limits the performance of RL algorithms. The bias may be introduced to the estimated value function based on TD learning due to the off-policy TD learning's max operation, chosen imperfect policy, and uncertainties (Sutton and Barto, 2018). TD learning method is an important estimation method for the value function and is widely used in existing RL algorithms, e.g., PPO, DDPG, etc. Especially for deterministic policy algorithms, e.g., DDPG, TD learning's max operation may lead to an overestimated Q-value (Van Hasselt et al., 2016), bringing negative effects to the policy update. Although the TD3 (Fujimoto et al., 2018) improves the overestimation, TD3 may lead to the underestimated Q-value and increase the complexity of the algorithm significantly. Additionally, as analyzed in Equation (7), sampling error caused by incomplete samplings can further increase the bias for the stochastic policy algorithms that rely on the estimated state-value function V_{π}^{E}, e.g., A3C, PPO, etc. Note that some complex scenario involving uncertain environments may generate completely different response results for the same state-action pair. Such complex and uncertain responses can bring a high variance for the estimated value functions, leading low reliability of controller. However, existing RL algorithms do not solve this problem very well.

As analyzed in Equation (13), a characteristic of the atypical MDPs is that they focus only on the maximum immediate return. And there is no focus on the long-term return. However, there are many designs for estimating long-term returns in existing RL algorithms. For example, based on Equations (2), (3), many existing RL algorithms, such as PPO, DDPG, etc., have an operation to calculate the successor state-value (or Q-value). When dealing with an atypical MDP case, such an operation is redundant and increases the time complexity of the algorithms, e.g., PPO. Especially for deterministic policy algorithms, e.g., DDPG and TD3, they contain a set of complex target networks to calculate the successor Q-value. It shall increase a great of both time complexity and space complexity. Such limitations can increase computing resource usage, which is not conducive to applying RL algorithms to complex problems.

Immediate-return RL algorithm for the atypical MDPs

The immediate-return RL algorithm

As analyzed in Section The atypical MDPs, deterministic policy shows more advantages than stochastic policy in atypical MDPs. Thus, the immediate-return RL algorithm is proposed based on the deterministic policy method and actor-critic framework for the problems in atypical MDPs. The new equations involved in this algorithm are highlighted in "⇐". As shown in Figure 1, two networks, i.e., actor network with weights θ^{μ} and critic network with weights θ^{Q}, are designed to construct the actor-critic framework. Here the actor network plays a role as the policy. It can output deterministic action $a_t = \mu\left(s_t\middle|\theta^{\mu}\right)$ based on the inputted state s_t. The critic network is used as the estimated action-value function. It can evaluate the performance of the actor network by outputted the estimated Q-value $Q\left(s_t,\ a_t\middle|\theta^{Q}\right)$ according to the inputted state-action pair $\left(s_t,\ a_t\right)$. Compared to other deterministic policy algorithms (e.g., DDPG), the proposed algorithm's network framework has been simplified significantly due to no target networks. It means less computing resource usage.

As analyzed in Equation (12), the true action-value function Q^{A} in atypical MDPs is equal to the expected reward $R(s_t, a_t)$. As shown in Equation (13), the immediate reward r_t (i.e., immediate return) is the unbiased estimation for the expected reward $R(s_t, a_t)$.

$$E_{r_t,s_{t+1}}\left[r_t\right] = \sum\nolimits_{s_{t+1},r_t} P\left(s_{t+1},r_t\middle|s_t,a_t\right)r_t = R(s_t,a_t) \Leftarrow (14)$$

When the environment is deterministic, the generated next state s_{t+1} and immediate reward r_t are also deterministic under

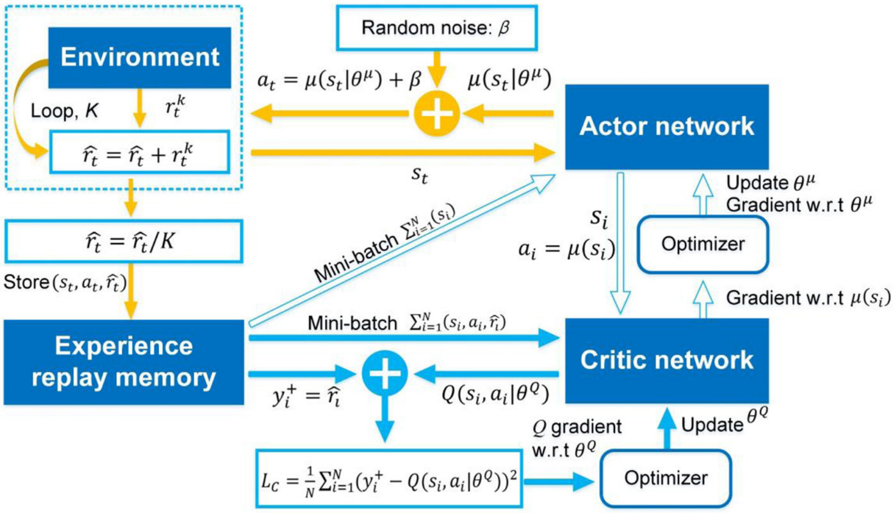

Environment
Loop, K
r_t^k
$\hat{r}_t = \hat{r}_t + r_t^k$
$\hat{r}_t = \hat{r}_t / K$
Store $(s_t, a_t, \hat{r}_t)$
Experience replay memory
Random noise: β
$a_t = \mu(s_t|\theta^\mu) + \beta$
$\mu(s_t|\theta^\mu)$
Actor network
s_t
Mini-batch $\Sigma_{i=1}^N(s_i)$
s_i
$a_i = \mu(s_i)$
Update θ^μ
Gradient w.r.t θ^μ
Optimizer
Gradient w.r.t $\mu(s_i)$
Mini-batch $\Sigma_{i=1}^N(s_i, a_i, \hat{r}_i)$
Critic network
$y_i^+ = \hat{r}_i$
$Q(s_i, a_i|\theta^Q)$
Q gradient w.r.t θ^Q
Update θ^Q
$L_C = \frac{1}{N}\Sigma_{i=1}^N(y_i^+ - Q(s_i, a_i|\theta^Q))^2$
Optimizer

FIGURE 1
The framework of the immediate-return RL algorithm. Yellow solid arrows: the actor network interacts with the environment. Blue solid arrow: update for critic network. Blue hollow arrow: update for actor network.

the specified state-action pair (s_t, a_t). Under this condition, the immediate reward r_t is equal to its expectation, i.e., $r_t = R(s_t, a_t)$. Thus, r_t is the ideal target Q-value y_t^+ of the estimated action-value function in an atypical MDP with a deterministic environment. However, the uncertain environments (e.g., dynamic systems with uncertainties) may generate different immediate rewards r_t even given the same state-action pair (s_t, a_t). A randomly generated reward value r_t cannot represent the expected reward $R(s_t, a_t)$ under the specified state-action pair (s_t, a_t). Due to the complexity of the uncertainties, the probability distribution of these generated reward values is also unknown. Therefore, using r_t as the critic network's target Q-value will result in a high estimation variance when considering uncertainties. The high variance may lead to instability in the learning process, making the policy less reliable (Fujimoto et al., 2018; Sutton and Barto, 2018). Based on the law of large numbers, the average reward $\hat{r}_t$ is proposed as the target Q-value to solve the problem of high variance caused by uncertain environments. $\hat{r}_t$ can be expressed as follows.

$$\hat{r}_t = \frac{1}{K}\sum\nolimits_{k=1}^{K} r_t^k \Leftarrow \tag{15}$$

That is, a specified state-action pair $(s_t,\ a_t)$ will be performed multiple times K in the uncertain environment. And a set including multiple immediate rewards $\left\{r_t^k\right\}$ will be obtained. This immediate reward set $\left\{r_t^k\right\}$ can reflect the probabilistic characteristics of the uncertain environment's responses under the state-action pair $(s_t,\ a_t)$. Then, the average reward $\hat{r}_t$ is constructed by averaging the immediate reward set $\left\{r_t^k\right\}$. According to the law of large numbers, the average reward $\hat{r}_t$ is closer to the expected reward $R(s_t, a_t)$ than one randomly generated reward r_t. Thus, there will be minor variance, when the average reward $\hat{r}_t$ is used to estimate the expected reward $R(s_t, a_t)$. Note that the average reward $\hat{r}_t$ is still the unbiased estimation for the expected reward $R(s_t, a_t)$ due to the average operation. In practical application, the repetition number K is suggested to be 3 based on experience. In the football trajectory control problems considered in this paper, setting the repetition number K to 3 can significantly improve the training results compared to setting the number of repetitions to 1. When K continues to increase, the algorithm's performance cannot be significantly improved. Based on the above analysis, these improvements provide an unbiased and low variance target Q-value for the critic network. It can make the proposed algorithm more reliable in uncertain environments. The problem of the estimation error in the existing RL algorithms, e.g., overestimation in DDPG, can also be overcome. The numerical tests in Section Controller's performance will prove this. Then, the new target Q-value y_t^+ of the immediate-return RL is expressed as

$$y_t^+ = \hat{r}_t \Leftarrow \tag{16}$$

It should be noted that the average reward $\hat{r}_t$ applies only to the atypical MDP case, as the successor state s_{t+1} relying on the current state-action pair (s_t, a_t) does not exist.

The off-policy method (Levine et al., 2020) is also adopted in the proposed algorithm. All training samples generated from the trial and error should be stored in the experience memory. It should be stressed that the atypical MDP case does not focus on the successor state s_{t+1}, and its trajectory contains only one state transition. Hence, only the initial samples of each trajectory, i.e., $(s_t,\ a_t,\ \hat{r}_t), t \equiv 0$, should be stored. Sampling N training samples $\sum_{i=1}^{N}(s_i,\ a_i, \hat{r}_i)$, the loss function for updating critic network is expressed as follows.

$$L_C = \frac{1}{N}\sum_{i=1}^{N}(y_i^+ - Q(s_i, a_i|\theta^Q))^2 \ \Big| y_i^+ = \hat{r}_i \tag{17}$$

where N is the size of the min-batch. Symbol i is the label number of the sample. By minimizing the loss function, the critic network weights θ^Q can be updated. Meanwhile, as analyzed in Equation (13), the policy should be updated in the direction of maximizing the expected reward. Thus, the purpose of updating the actor network μ is to maximize the estimated Q-value outputted by the critic network. Referring to Lillicrap et al. (2015), the gradient for updating actor network weights θ^μ is expressed as follows.

$$\nabla_{\theta^\mu}|_{s_i} = \frac{1}{N}\sum_{i=1}^{N}\nabla_{a_i}Q\left(s_i, a_i = \mu(s_i)\middle|\theta^Q\right)\ \nabla_{\theta^\mu}\mu\left(s_i|\theta^\mu\right) \tag{18}$$

Furthermore, the delaying policy update (Fujimoto et al., 2018) is also introduced for the immediate-return RL algorithm. It can reduce the frequent policy updates and further result in low variance (Fujimoto et al., 2018). After successful training, the actor network will be the RL controller.

In summary, due to the proposed average reward method, the open problem of the estimation error can be improved significantly in the proposed algorithm. Compared to existing RL algorithms, the proposed algorithm will show high performance. This point will be certified in two football trajectory control scenarios (see Section Comparison and discussion). Besides, based on the characteristics of atypical MDPs, a simplified network framework is designed for the proposed algorithm to reduce computing resource usage. Then, the complete pseudocode of the immediate-return RL algorithm is shown in Algorithm 1.

Complexity analysis

The computing complexity, i.e., space complexity and time complexity, can reflect the requirement of the algorithm for computing resources. To verify the low computing resource requirement of the immediate-return RL algorithm, the computing complexity of the proposed algorithm will be analyzed in this section. Meanwhile, the representative of the stochastic policy algorithms, i.e., PPO, and the representative of the deterministic policy algorithms, i.e., DDPG, will also be analyzed as references. For the details of DDPG and PPO, please see Lillicrap et al. (2015), Schulman et al. (2017). In the following analysis, the single network's detailed architectures in Section Training process will be used as an example for clarity.

```
Randomly initialize actor network $\mu$ with weights $\theta^\mu$
Randomly initialize critic network $Q$ with weights $\theta^Q$
Initialize the experience replay memory $E$
For step $t$= 1, T do
    Generate initial state $s_t$ in the environment
    Output action $a_t = \mu(s_t|\theta^\mu) + \beta$ based on current policy and random noise
    Initialize average reward $\hat{r}_t = 0$
    For $k$= 1, $K$ do
        Running the state-action pair $(s_t, a_t)$ in environment on the $kth$ times
        Observe the reward $r_t^k$, and $\hat{r}_t = \hat{r}_t + r_t^k \Leftarrow$
    End Loop $K$
    Calculate average reward $\hat{r}_t = \hat{r}_t/K \Leftarrow$
    Store the sample $(s_t,\ a_t,\ \hat{r}_t)$ in $E$
    Extract random a minibatch of $N$ samples $\sum_{i=1}^{N}(s_i,\ a_i, \hat{r}_i)$ from $E$
    Obtain the target Q-value $y_i^+ = \hat{r}_i \Leftarrow$
    Construct the loss function $L_C$ of the critic network:
    $L_C = \frac{1}{N}\sum_{i=1}^{N}(y_i^+ - Q(s_i, a_i|\theta^Q))^2$
    Update the critic network weights $\theta^Q$ by minimizing the loss $L_C$
    If $t$ mod $d$ then
        Update the actor network weights $\theta^\mu$ using the gradient:
        $\nabla_{\theta^\mu}|_{s_i} = \frac{1}{N}\sum_{i=1}^{N}\nabla_{a_i}Q\left(s_i, a_i = \mu(s_i)\middle|\theta^Q\right)\ \nabla_{\theta^\mu}\mu\left(s_i|\theta^\mu\right)$
    End IF
End Loop $T$
```

Algorithm 1 The immediate-return RL algorithm.

Since the algorithms mentioned above are composed of networks, their space complexity depends on the total parameter of all networks. According to Han et al. (2015), the whole space complexity of a single network is:

$$space \sim O\left(\sum_{l=1}^{L-1} N_l N_{l+1} + N_{l+1}\right) \tag{19}$$

where L=5 is the total layer number of the networks. N_l represents the total node number of the l layer. As shown in Table 1, both the proposed algorithm and PPO have two networks (Schulman et al., 2017), and the DDPG contains four networks (Lillicrap et al., 2015). Thus, the space complexity of the proposed algorithm is similar to PPO and is reduced by 50% compared to DDPG.

The time complexity of the RL algorithms depends on both the network framework and the calculation process (i.e., sampling process and update process). Generally, floating point operations (FLOPs) is used to evaluate the algorithm's time complexity. Referring to He and Sun (2015), the time complexity of a single network is:

$$time \sim O(\sum_{l=1}^{L-1} 2N_l N_{l+1}) \tag{20}$$

TABLE 1 The space and time complexity analysis.

		The proposed algorithm	DDPG	PPO
Space complexity	Actor network	199,558	199,558	200,332
	Critic network	200,449	200,449	198,913
	Target networks	\	400,007	\
	Total	400,007	800,014	399,245
Time complexity (FLOPs)	Actor network	397,312	397,312	398,848
	Critic network	399,104	399,104	396,032
	Target networks	\	796,416	\
	Once Sampling	397,312	397,312	398,848
	Once network Update	796,416	1,592,832	(794,880∼1,190,912)

Then, the time complexity of one sampling and one network update will be discussed separately (see Table 1). For the three algorithms discussed in this article, only the actor network is working when sampling. Thus, the time complexity of the three discussed algorithms can be regarded as the same in one sampling and is equal to the actor network's time complexity (see Table 1). Note that although the state-action pair (s_t, a_t) will be performed many times in the environment (Algorithm 1 Line 8 to Line 10), the proposed algorithm's time complexity will not be increased in one sampling, as its actor network only runs once. Regarding the time complexity of network updates, only the network's forward computation is considered according to He and Sun (2015). When the proposed algorithm and DDPG (Lillicrap et al., 2015) update their networks, all their networks will be used once. Here, the proposed algorithm has 2 networks, and DDPG has four (Lillicrap et al., 2015). Thus, the proposed algorithm reduces the time complexity of each network update by 50% than DDPG (see Table 1). In each network update, the actor network and critic network of PPO should estimate $\pi(s_t)$ the $V(s_t)$, respectively (Schulman et al., 2017). Besides, for the same batch of samples that are trained multiple times, the critic network should be used once to estimate $V(s_{t+1})$ due to the TD learning method (Schulman et al., 2017). That is, the fewer times the same batch of samples are trained, the greater the time complexity of each network update. When a batch of samples is used only once, the proposed algorithm can reduce the time complexity of each network update by 33.1% than the PPO (see Table 1). Thus, based on the above analysis, when the sampling times and the network update times are constants, the time complexity of the proposed algorithm is 40% lower than the DDPG and 0–24.9% lower than the PPO.

It should be stressed that computing resources are limited and precious. Especially for some actual complex tasks involving vision, the usage of computing resources is enormous. Based on the above analysis, the immediate-return RL algorithm has lower computing complexity than the existing RL algorithms, reducing computing resource usage. Such statements will be verified in the following Section Computing resource usage by detailed comparisons.

Illustration examples: Football trajectory control for different scenarios

The football flight is an atypical MDP case. To test the immediate-return RL algorithm, two highly challenging scenarios involving the flight control of the football, i.e., passing the football to a moving player, and chipping the football over the human wall, will be examined. These scenarios can be used as the benchmark to test the algorithms designed for the atypical MDPs. Meanwhile, regarding research results can be used to develop high-level football robots in the Robot world cup (Sharbafi et al., 2011). The controllers based on the proposed algorithm in this paper can significantly increase the accuracy of the football shot.

Under the above two scenarios, the proposed controllers will be trained to output accurate initial velocities for the football to achieve the specified flight purposes and reduce the time of football flight. In the following sections, the experimental model will be introduced in Section Experimental model: Aerodynamic model of football with parameter uncertainties first. Then, other detailed designs corresponding to the two different scenarios, including the actions designs, states designs and constraints, the termination events definitions, and the reward function designs, will be introduced in Section Scenario 1: passing the football to a moving player and Section Scenario 2: chipping the football over the human wall.

Experimental model: Aerodynamic model of football with parameter uncertainties

Here, an aerodynamic model of football under windless conditions is directly reproduced here from Myers and Mitchell

TABLE 2 The fitting coefficients of the drag coefficient function (Kiratidis and Leinweber, 2018).

Balls	a_c	v_c	v_s	b_{min}	b_{max}	v_{min}	v_{max}	b_r
Tango12	0.5452	12.8600	1.3040	0.1657	0.1953	16.2200	35.0000	0.5332
Teamgeist	0.4927	12.5800	1.0710	0.1440	0.1540	23.1700	35.0000	0.5140
Brazuca	0.4740	12.9200	1.0000	0.1657	0.2112	14.6100	35.0000	0.5397

These fitting coefficients are derived from the actual wind tunnel data of famous footballs, including Tango12, Teamgeist and Brazuca.

(2013), Javorova and Ivanov (2018). On this basis, parameter uncertainties are newly introduced into the aerodynamic model of the football. Thus, the football flight process can be regarded as an uncertain environment. This aerodynamic model will be adopted directly as the simulation environment to further generate the training data for the RL controllers.

The external forces acting on the ball include gravity $\mathbf{G}$, drag force $\mathbf{F}_D$, lift force F_L, and drag moment $\mathbf{M}_D$. Thus, the aerodynamic model of football can be expressed as follows (Myers and Mitchell, 2013; Javorova and Ivanov, 2018).

$$\tilde{m}\ddot{x} = -K_D\dot{x}\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2} + K_L\left(\dot{x}^2+\dot{y}^2+\dot{z}^2\right)(\omega_Y\dot{z}-\omega_Z\dot{y}) \tag{21}$$

$$\tilde{m}\ddot{y} = -K_D\dot{y}\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2} + K_L\left(\dot{x}^2+\dot{y}^2+\dot{z}^2\right)(\omega_Z\dot{x}-\omega_X\dot{z}) \tag{22}$$

$$\tilde{m}\ddot{z} = -K_D\dot{z}\sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2} + K_L\left(\dot{x}^2+\dot{y}^2+\dot{z}^2\right)(\omega_X\dot{y}-\omega_Y\dot{x}) - \tilde{m}\mathbf{g} \tag{23}$$

$$\dot{\omega}_X = -\eta\omega_X \tag{24}$$

$$\dot{\omega}_Y = -\eta\omega_Y \tag{25}$$

$$\dot{\omega}_Z = -\eta\omega_Z \tag{26}$$

where parameters K_D and K_L are specified as follows.

$$K_D = 0.5\tilde{C_d}\tilde{\rho}\pi\tilde{r}^2 \tag{27}$$

$$K_L = 0.5C_L\tilde{\rho}\pi\tilde{r}^2\frac{1}{|\omega\times\mathbf{v}|} \tag{28}$$

here, $\tilde{m}$ is the football's mass, $\mathbf{g}$ is the gravitational acceleration, $\tilde{\rho}$ is the air density, $\tilde{r}$ is the radius of the football, $\mathbf{v}$=$(\dot{x},\dot{y},\dot{z})$ is the linear velocity, and ω= $(\omega_X,\omega_Z,\omega_Y)$ is the angular velocity. The attenuation coefficient η is assumed to be 0.05. Furthermore, the dimensionless lift coefficient C_L is adopted from Kiratidis and Leinweber (2018) as follows.

$$C_L = \left(1-\partial v^2\right)S_p{}^{\beta} \tag{29}$$

here, the parameter ∂ is chosen as 2.5×10^{-4}, and β is 0.83 (Kiratidis and Leinweber, 2018). The spin parameter is $S_p = \frac{\tilde{r}\omega}{v}$, where $\omega = |\omega|$ and $v = |\mathbf{v}|$. The dimensionless drag coefficient is expressed as $\tilde{C_d}$, which is an important factor for the sudden change of linear velocity of football in flight (Horowitz and Williamson, 2010; Norman and McKeon, 2011). Its fitting function is adopted from Kiratidis and Leinweber (2018) as follows.

TABLE 3 The range of the uncertain parameters.

Uncertainparameters	Unit	Minimum value	Maximum value
Air density $\tilde{\rho}$	kg/m^3	1.000	1.205
Mass $\tilde{m}$	kg	0.42	0.45
Radius $\tilde{r}$	m	0.1090	0.1106

$$\tilde{C_d}\left(v,sp\right) = \frac{a_c-b_{\min}}{1+e^{\frac{v-v_c}{v_s}}} + b_{\min} + \frac{v-v_{\min}}{1+e^{\frac{-v+v_{\min}}{v_s}}}\frac{b_{\max}-b_{\min}}{v_{\max}-v_{\min}} + b_rS_p \tag{30}$$

where a_c, $b_{\min}$, $b_{\max}$, b_r, $v_{\min}$, $v_{\max}$, v_c, and v_s, are the fitting coefficients of the above function (see Table 2).

Next, parameter uncertainties, i.e., air density $\tilde{\rho}$, mass $\tilde{m}$, radius $\tilde{r}$, and drag coefficient $\tilde{C_d}$, in the aerodynamic model of football will be introduced. Here, $\tilde{m}$, $\tilde{\rho}$, $\tilde{r}$, and $\tilde{C_d}$, are internal parameters, and $\tilde{\rho}$ is external parameter. All parameters with uncertainties are random and parametric. The following parameters, i.e., $\tilde{\rho}$, $\tilde{m}$, and $\tilde{r}$, can change in very small intervals according to the international federation of association football (FIFA) standards, and these details are shown in Table 3. In addition, the different fitting coefficients of the drag coefficient functions (Kiratidis and Leinweber, 2018) corresponding to three kinds of footballs, i.e., Tango12, Teamgeist, or Brazuca, are considered in this paper (see Table 2). That is, when giving specified initial conditions and simulating Equations (21)–(26) in the training or testing procedures, values of the $\tilde{\rho}$, $\tilde{m}$, and $\tilde{r}$ will be selected randomly from Table 3, and one set of the fitting coefficients of the drag coefficient function will be selected randomly from the Table 2. Note that slight changes in the above parameters can significantly impact the flight trajectories, although the football has the same initial condition. In order to analyze the impact of the parameter uncertainties, 20 random initial conditions are generated to test. Based on Equations (21)–(26), each initial condition is simulated 100 times and produces 100 flight trajectories. In each initial condition, the average landing position of these 100 flight trajectories is set as the target position. Then, the average relative error of the 100 landing positions relative to the target position can be calculated to assess

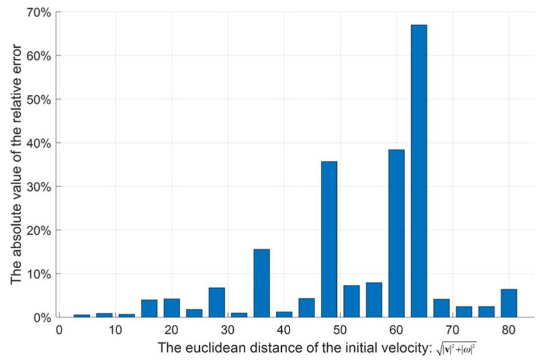


FIGURE 2
The effect of parameter uncertainty on flight trajectory.

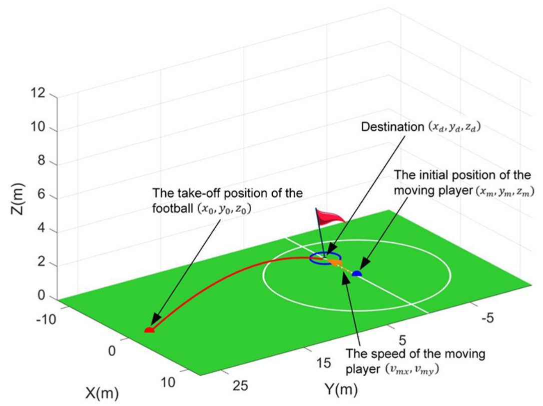

12
10
8
6
4
2
0
Z(m)
-10
0
10
X(m)
25
15
5
-5
Y(m)
Destination (x_d, y_d, z_d)
The initial position of the moving player (x_m, y_m, z_m)
The take-off position of the football (x_0, y_0, z_0)
The speed of the moving player (v_{mx}, v_{my})

FIGURE 3
Passing the football to a moving player.

the impact of the parameter uncertainties. The results of 20 tests are shown in Figure 2. Here, the maximum average relative error is 66.97%. The average value of 20 average relative errors is 10.63%. Thus, parameter uncertainties can have a non-negligible impact on the flight trajectory and pose a significant challenge to the controller design.

Scenario 1: Passing the football to a moving player

The schematic diagram of the first scenario is shown in Figure 3. And this scenario simulates the dynamic passing situation between the players in reality. That is, the moving player moves when the football flies and stops when the football lands. Here, two control targets, i.e., passing the football to a moving player and reducing the time of the football flight, are set for the RL controller.

The action outputted by the RL controller is the initial velocities of the football, i.e., initial linear velocity and initial angular velocity. This action is designed as follows.

$$A_0 = \left(v_x, v_y, v_z, \ \omega_x, \omega_y, \omega_z\right) \tag{31}$$

It should be noted that both the linear and angular velocities should be limited according to the practical data of the professional players (Neilson, 2003), i.e., $|\mathbf{v}| \in [0, \ 34]$ m/s and $|\omega| \in [0, \ 62.8]$ rad/s.

In this scenario, the initial position of the moving player will be set at the coordinate origin for convenience, i.e., $(x_m, y_m, z_m) = (0, \ 0, \ 0)$. Thus, the conditions when the football takes off, i.e., state S_1, can be described as follows.

$$S_1 = \left(x_0, y_0, z_0, v_{mx}, v_{my}\right) \tag{32}$$

where (x_0, y_0, z_0) is the football's initial take-off position. The (v_{mx}, v_{my}) is the moving speed of the moving player. Then, the constraints for the state S_1 are set as follows. Firstly, according to the player's sprint speed (Djaoui et al., 2017), the maximum speed of moving players is limited to 10 m/s, i.e.,

$$v_m = \sqrt{{v_{mx}}^2 + {v_{my}}^2} \leq 10 \tag{33}$$

Secondly, considering the size of the sports field, the constraint on the choice of the take-off position is defined as follows.

$$d_m = \sqrt{(x_0 - x_m)^2 + (y_0 - y_m)^2 + (z_0 - z_m)^2} \leq 30 \tag{34}$$

Note that the destination $\left(x_d, y_d, z_d\right)$ of the football in this scenario is defined as the end position of the moving player, i.e., $(x_m + v_{mx}t_f, y_m + v_{my}t_f, z_m)$. t_f is the football's flight time. That is, the destination is not a constant pre-defined in the state S_1 and unknown for the RL controller. Thus, passing the football to a moving player is a challenging scenario.

To generate reasonable trajectories, some termination events of the simulations should be set according to the constraints required. Any of termination events are triggered, the flight process will be stopped. In this scenario, the ground floor $Z_{LB} = 0$ and maximum height $Z_{LH} = 12$ are set as the constraints for flight trajectories. Therefore, the termination events for this scenario are defined as $z_f = Z_{LB}$ or $z_f = Z_{LH}$. Here, the $\left(x_f, y_f, z_f\right)$ denotes the football's final position when the termination event is triggered.

For the purpose of learning an excellent policy to predict proper initial velocities, the RL controller needs to be guided by an appropriate reward function. Here, a monotonic power function (i.e., $y = 1 - x^b$) is selected as the basic function to design the reward function. For this basic function, the closer x is to 0, the greater the change in the gradient $\frac{dy}{dx}$. Thus, the reward function based on this power function can provide very large positive rewards for a small number of correct samples in some complex scenarios. It may provide more precise guidance for RL algorithms. Note that other function forms may also have similar effects, and the proposed basic functions only offer an effective solution. In this scenario, two sub-reward functions based on this basic function are designed for two independent control targets, i.e., passing the football to a moving player, and reducing the time of football flight, respectively. Then, two sub-reward functions will be combined into one united reward function by reward shaping (Brys et al., 2017) to integrate these two control targets.

For the first control target, i.e., passing the football to a moving player, the relative error δ between the football's final position $\left(x_f, y_f, z_f\right)$ and the destination $\left(x_d, y_d, z_d\right)$ is a reasonable parameter to evaluate the flight results. It can be expressed as follows,

$$\delta = \Delta d / d_d \tag{35}$$

where, Δd is the absolute error between the football's final position and the destination, i.e.,

$$\Delta d = \sqrt{\left(x_d - x_f\right)^2 + \left(y_d - y_f\right)^2 + \left(z_d - z_f\right)^2} \tag{36}$$

d_d indicates the distance between the take-off position and the destination, i.e.,

$$d_d = \sqrt{\left(x_0 - x_d\right)^2 + \left(y_0 - y_d\right)^2 + \left(z_0 - z_d\right)^2} \tag{37}$$

Then, the first sub-reward function is designed as follows,

$$r_{1,1} = 1 - \delta^{0.4} \tag{38}$$

where constant-coefficient 0.4 is an empirical parameter by error and trial. For the sub-reward function $r_{1,1}$, the smaller the relative error, the faster the reward increases. This character will benefit the convergence of the networks in the proposed algorithm. For the second control target, i.e., reducing the time of football flight, the unit time cost index t_s is defined as follows.

$$t_s = t_f / d_m \tag{39}$$

where t_f is the football's flight time and parameter d_m can be found in Equation (34). Then, the second sub-reward function is defined as follows:

$$r_{1,2} = 1 - \left(\max(t_s - t_0, 0)\right)^{0.15} \tag{40}$$

where symbol $t_0 = 0.055\ s/m$ is the empirical value based on simulations, which indicates the expected unit time cost. As defined by the sub-reward function $r_{1,2}$, the lower the unit time cost index, the higher the value of the reward. Then, the united reward function will be shaped as Equation (41).

$$R_1 = \frac{14}{9} r_{1,1} + \frac{4}{9} r_{1,2,} \; z_f = Z_{LB}\ or\ z_f = Z_{LH} \tag{41}$$

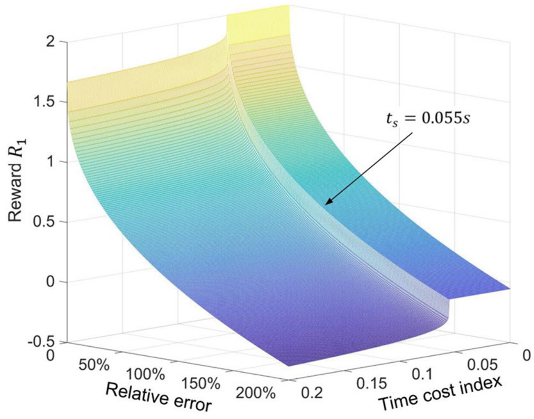


FIGURE 4
Distribution of the value of the reward function R_1 on the parameter t_s and parameter δ.

where the value of the reward function R_1 is restricted from 0 to 2 according to the recommended value of the Henderson et al. (2018). Considering the importance of the first control target and the value limitation of the R_1, $\frac{14}{9}$ and $\frac{4}{9}$ are selected the shaping weights for $r_{1,1}$ and $r_{1,2}$, respectively. Since the different control targets have different sensitivities in reward value, reasonable shape weights are helpful to find the optimal policy that can satisfy multiple control targets. However, these weights in reward shaping usually originate in practical experience. The pretest results also demonstrate that changing the shaping weights value will decrease the proposed controllers' performance. After shaping, the distribution of the reward function R_1 on relative error δ and unit time cost index t_s is shown in Figure 4.

Scenario 2: Chipping the football over the human wall

The schematic diagram of chipping the football over the human wall is shown in Figure 5. In this scenario, the football is required to fly over (rather than through) the human wall and reach at the goal. Indeed, this scenario simulates the free kick situation in the football game. Similar as the first scenario, the action outputted by the RL controller is the football's initial velocities, which are defined in Equation (31). Here, two control targets, i.e., chipping the football into the goal and reducing the time of the football flight, are set for the RL controller.

In this scenario, the goal is defined as perpendicular to the positive Y-axis and the projection of the goal's central point on the X-Y plane is set at the coordinate origin $(0, 0, 0)$. Thus, the central point of the goal will be always regarded as the destination, i.e., $\left(x_d, y_d, z_d\right) = (0, 0, 1.22)$ (The height of the goal

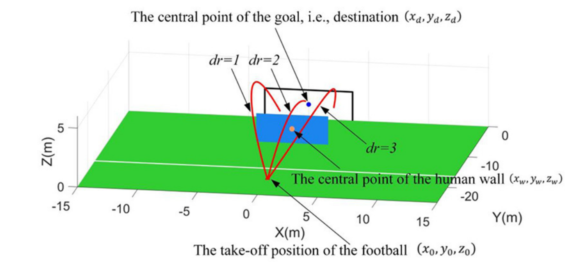


FIGURE 5
Chipping the football over the human wall.

is 2.44 m based on the FIFA standards). Then, a 2.4 × 6 m human wall parallel to the goal is placed between the goal and take-off position of football. Here, the projection points of the human wall's central point, the goal's central point, and the football's take-off position on the X-Y plane are assumed to be collinear. Thus, the conditions when the football takes off in this scenario, i.e., state S_2, can be described as follows.

$$S_2 = \left(x_0, y_0, z_0, dr, x_w, y_w, z_w\right) \tag{42}$$

where, the $\left(x_0, y_0, z_0\right)$ can be found in Equation (32). The $\left(x_w, y_w, z_w\right)$ represents the central point of the human wall. The parameter dr represents the specified direction requirement for flight trajectories. Namely, $dr = 1$ is left side of the human wall, $dr = 2$ is top of the human wall, and $dr = 3$ is right side of the human wall. Then, the constraints for the state S_2 are defined as follows. The constraints for take-off position are set as $x_0 \in [-20, 20]$ and $y_0 \in [-15, -25]$. Note that $z_0 \equiv 0$. Based on the free-kick rules, the constraint for the human wall's position is defined as follows,

$$\sqrt{(x_0 - x_w)^2 + (y_0 - y_w)^2} \geq 9.15 \tag{43}$$

Due to the human wall, the flight trajectories of footballs are required to specified shapes. Meanwhile, multiple specified direction requirements are considered, which means more functional requirements. Thus, the complexity of this scenario is significantly increased more than the first scenario.

In this scenario, another two termination events should be defined, besides two termination events $z_f = Z_{LB}$ or $z_f = Z_{LH}$ described in Section Scenario 1: Passing the football to a moving player. Here, the third termination event triggered by the human wall ($y_f = y_w$) is required. That is, the football bumps into the human wall. Based on the parameter dr, the third termination event has three triggering conditions, which can be expressed as follows.

$$\begin{cases} x_f \geq x_w - 3\text{ , when dr} = 1 \text{ and } \mathrm{y_f} = \mathrm{y_w} \\ \left|x_f - x_w\right| > 3 \; or \; z_f \leq 2z_w\text{, when dr} = 2 \text{ and } \mathrm{y_f} = \mathrm{y_w} \\ x_f \leq x_w + 3,\text{ when dr} = 3 \text{ and } \mathrm{y_f} = \mathrm{y_w} \end{cases} \tag{44}$$

Then, the fourth termination event indicates that the football reaches at the two-dimensional surface corresponding to the goal, which is written as $y_f = 0$.

Since the complexity of the control requirements in the second scenario, three independent reward functions, i.e., $R_{2,1}$, $R_{2,2}$, and $R_{2,3}$, are designed respectively depending on the triggering four termination events. Note that triggering the fourth termination event $y_f = 0$ is the essential precondition for chipping the football into the goal. Thus, only when the fourth termination event is triggered, reducing the time of football flight should be considered, and the relevant reward function $R_{2,3}$ is set from 0 to 2. And other reward functions $R_{2,1}$ and $R_{2,2}$ are defined between −2 and 0 to ensure the coherence of the reward's guidance (see Figure 6). Since each reward function only works on a specified termination event, a simple linear function (i.e., $y = kx + b$) is also selected as the reward's basic function besides the power function.

When the first and second termination events are triggered, i.e., $z_f = Z_{LB}$ or $z_f = Z_{LH}$, the first reward function is designed as follows to guide the football close to the destination.

$$R_{2,1} = -2\delta,\ z_f = Z_{LB} \; or \; z_f = Z_{LH} \tag{45}$$

where δ can be found in Equation (35). When the third termination event takes effect, that is, the football hits the human wall, the second reward function should guide the ball to fly over the human wall. Based on the definition of the third termination event's triggering conditions in Equation (44), the second reward function can be expressed as follows.

$$r_{2,2} = \begin{cases} -0.17\left(x_f - x_w + 3\right), & dr = 1,\ y_f = y_w, x_f \geq x_w - 3 \\ -0.17\left(\left|x_w - x_f\right| - 3\right) - 0.5, & dr = 2,\ y_f = y_w, \left|x_f - x_w\right| > 3 \\ -0.21\left(2z_w - z_f\right), & dr = 2,\ y_f = y_w, z_f \leq 2z_w \\ -0.17\left(x_w + 3 - x_f\right), & dr = 3,\ y_f = y_w, x_f \leq x_w + 3 \end{cases} \tag{46}$$

When the fourth termination event is triggered $y_f = 0$, two independent sub-reward functions are designed for chipping the football into the goal and reducing the time of the football flight, respectively. The first sub-function $r_{2,3a}$ is used to guide the football toward the goal, which is designed as follows.

$$r_{2,3a} = \begin{cases} -0.068\left|x_f - x_d\right| + 0.75, & y_f = 0,\ \left|x_f - x_d\right| > 3.66 \\ -0.14\left(z_f - z_d\right) + 1.1708, & y_f = 0,\ z_f - z_d > 1.22 \\ -0.26d + 3, & y_f = 0,\ else \end{cases} \tag{47}$$

here d can be found in Equation (36). The second sub-function $r_{2,3b}$ is used to optimize the flight time. Referring to the Equation (40), it can be expressed as follows.

$$r_{2,3b} = 1 - (\max(t_s - t_0, 0))^{0.15}, y_f = 0 \tag{48}$$

where the unit time cost index t_s is defined as $t_s = t_f/d_d$. The d_d can be found in Equation (37). And t_0 can be found

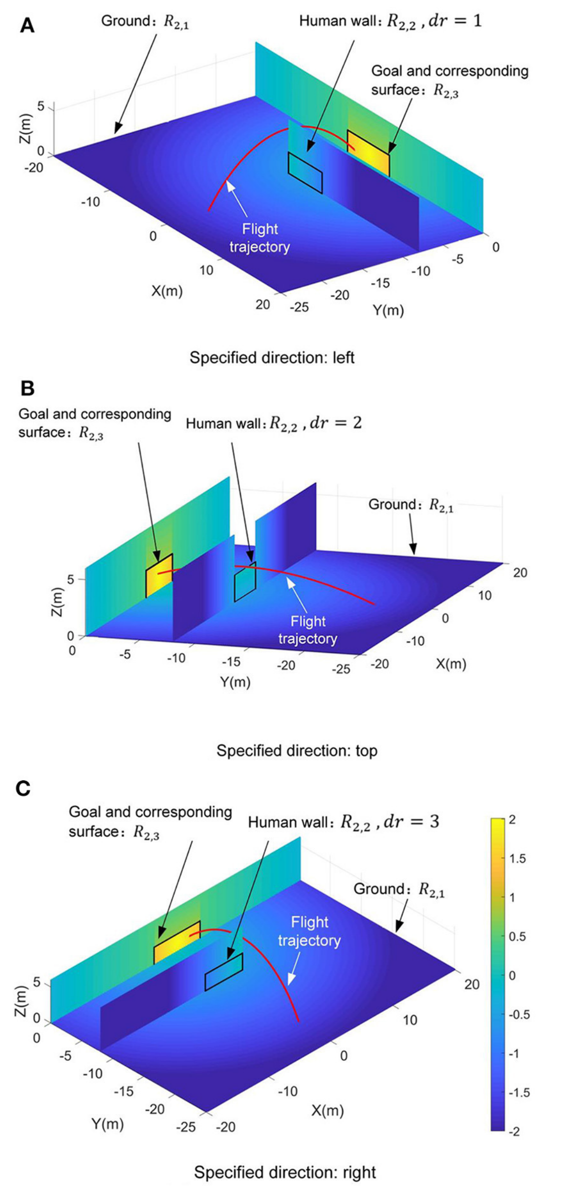


FIGURE 6
Distributions of the value of the reward function R_2 on the sports field. **(A)** The specified direction *dr* = 1. **(B)** The specified direction *dr* = 2. **(C)** The specified direction *dr* = 3.

in Equation (40). Then, the third reward function are shaped as Equation (49).

$$R_{2,3} = \frac{1}{2} r_{2,3a} + \frac{1}{2} r_{2,3b} \text{ , } y_f = 0 \tag{49}$$

where $\frac{1}{2}$ and $\frac{1}{2}$ are the shaping weights. Note that the value of the third reward function is designed to be larger than the first and second. This design can effectively guide the football reaching to the goal. Under the requirements of three specified directions, the distributions of the value of the reward function R_2 on the sports field are shown in Figure 6. Actually, reward function design is an experienced-based work (Dewey, 2014; Henderson et al., 2018; Silver et al., 2021). The constant-coefficients of the Equation (38), Equation (40), Equation (41), and Equations (45-49) are all determined by error and trial. And the pretest results verify that the proposed reward functions have strong guidance for optimizing control strategy under the effects of these constant-coefficients.

Comparison and discussion

In this section, the advantages of the immediate-return RL algorithm for atypical MDPs will be discussed and demonstrated. Meanwhile, PPO (the representative of the stochastic policy algorithms) and DDPG (the representative of the deterministic policy algorithms) are chosen as the references for the proposed algorithm. All these algorithms will train corresponding controllers for two football flight scenarios. Then, the advantages of the proposed algorithm will be discussed and analyzed from the training process, training results (i.e., the performance of the controllers), and computing resource usage by comparing with these reference algorithms.

Training process

For the control problems of the football trajectory, the proposed algorithm's detailed network framework is designed in Figure 7, including an independent actor network and an independent critic network. Here, the proposed algorithm's actor network and critic network have the same hidden layers and node numbers, i.e., the same network architectures. Indeed, each independent network in the three discussed algorithms shares the same network architectures to avoid the influence of the network architectures on the test results. Similarly, all discussed algorithms use the same reward function designed in Section Illustration examples: Football trajectory control for different scenarios. Furthermore, it should be noted that different deep RL algorithms have different sensitivities to hyperparameters (Henderson et al., 2018). Based on the trial and error and the experience of Dewey (2014), Henderson et al. (2018); and Silver et al. (2021), the detailed hyperparameters of each algorithm are selected (see Table 4). Under the premise of ensuring the algorithm's performance, each algorithm's hyperparameters are set to the same value.

Then, all algorithms, i.e., the proposed algorithm, DDPG, and PPO, will train the corresponding controllers for these two scenarios. Here, the learning efficiency of the algorithm can be evaluated by the consumption of the training steps. After 450,000 training steps, all reward curves in these two scenarios are shown in Figure 8. In both scenarios, the reward curves of the proposed algorithm (red line in Figure 8) converge to the

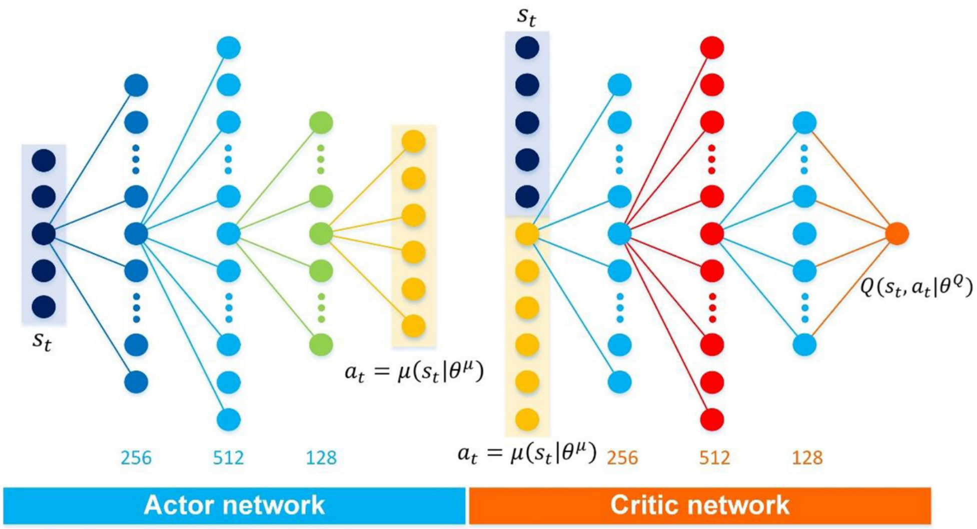


FIGURE 7
Detailed network architectures of the proposed algorithm.

TABLE 4 The hyperparameters of the discussed deep RL algorithms.

	Learning rate for actor	Learning rate for critic	Discount factor	Soft target updates
The proposed algorithm	1e-4	1e-4	\	\
DDPG	1e-4	1e-4	0.9	0.01
PPO	5e-6	1e-5	0.9	\

high-level reward value after 300,000 training steps. Then, the suitable controllers can be obtained. Although the reward value of the DDPG algorithm also has risen during training (green line in Figure 8), DDPG's learning efficiency is worse than the proposed algorithm from the perspective of convergence speed. As shown in Figure 8, DDPG needs about 450,000 training steps to converge the reward curves. That is, the learning efficiency of the proposed algorithm is 1.5 times that of the DDPG. And the convergency reward value of the DDPG is also less than the proposed algorithm. As a stochastic policy algorithm, PPO shows poor learning ability in football trajectory control. As show in Figure 8, 450,000 training steps do not allow the PPO to converge. Actually, PPO can also be converged after consuming about 1,500,000 training steps. That is, the learning efficiency of the proposed algorithm is 5 times that of the PPO. Furthermore, the final convergency reward values of the PPO are far less than the proposed algorithm. Note that the more training steps, the more samples are required. Thus, the training process confirms the analysis in Section Atypical MDPs: Definition and characteristic analyses. That is, PPO's learning efficiency is low in the atypical MDPs, as estimating a state-value requires more samples. The above training process demonstrates that the proposed algorithm converges faster and consumes fewer samples compared to DDPG and PPO. That is, the proposed algorithm shows better learning efficiency. Actually, it is a significant advantage for the proposed algorithm, as the samples are difficult to obtain in many atypical MDP cases.

Controller's performance

In this section, the performance of the controllers will be analyzed from three aspects, i.e., accuracy, unit time cost, and reliability. As described in Section Illustration examples: Football trajectory control for different scenarios, two control targets, i.e., shooting the football to the destination and reducing the time of football flight, are considered for each scenario. Thus, accuracy and unit time cost are the core index for evaluating the control performance of the controllers. Actually, the control performance is closely related to the value function's estimation bias. Besides, the considered aerodynamic model of football is an uncertain environment. That is, the football trajectory may be completely different under the same state-action pair, bringing a high variance for the value function. To evaluate the effect of variance caused by the uncertain environment on

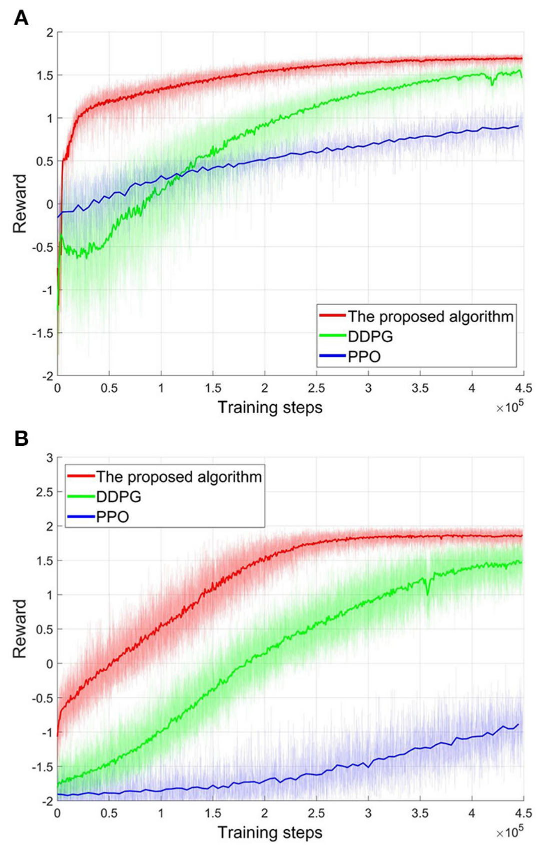


FIGURE 8
The reward curves of different algorithms. **(A)** The first scenario. **(B)** The second scenario.

the controller, the reliability is set as another index for the controller's performance.

Here, Monte Carlo tests are applied to analyze the control performance of the controllers. In each scenario, 1,000 independent state will be chosen randomly and a set of initial velocities will then be generated by the tested controller for each chosen state. Then, only one flight trajectory will be generated for the chosen state and the outputted initial velocities. Here, the effective rate of control Re is defined as follows to evaluate the accuracy of the RL controller.

$$Re = N_{Re}/1000 \qquad (50)$$

where N_{Re} is the number of the flight trajectories successfully controlled in 1000 tests.

For the first passing scenario, if the relative error δ is less than 5%, the flight control will be regarded as success. Here, the relative error δ is defined as follows.

$$\delta = \frac{\sqrt{(x_d - x_f)^2 + (y_d - y_f)^2}}{\sqrt{(x_d - x_0)^2 + (y_d - y_0)^2}} \qquad (51)$$

As shown in Figure 9A, the effective rate of control Re of the proposed controller in the first scenario, i.e., passing the football to a moving player, is 98.2%. In particular, there are 36.0% tests with relative error less than 1%, 56.6% tests with relative error from 1 to 3%, and 5.6% tests with relative error between 3 and 5%. Under the same tests, the DDPG controller's Re is 79.3%, and the PPO controller's Re is 80.5%. For the second scenario, scoring goals are regarded as the successful controls. The effective rate of control Re of the proposed controller for chipping the football over the human wall is 97.7% (see Figure 9B). Meanwhile, the DDPG controller's Re and PPO controller's Re are 91.1 and 24.1%, respectively. Compared to DDPG and PPO, the good accuracy of the proposed controller is verified in both two scenarios.

Based on 1,000 Monte Carlo tests, the average unit time cost t_a of 1,000 tests is used to evaluate the unit time cost, which can be written as.

$$t_a = \sum\nolimits_{1}^{1000} t_s/1000 \qquad (52)$$

here, t_s is the unit time cost index, which can be found in Equation (39). For the sake of comparison and evaluation, the proposed controllers without the time cost optimization are also trained for two scenarios. In the first scenario, the proposed controller reduces the average unit time cost t_a from 0.2080s to 0.0483s, comparing to the proposed controller without the time cost optimization (see Figure 10). Meanwhile, the DDPG controller can reduces the unit time cost t_a to 0.0484s. And the PPO controller can reduce the unit time cost t_a to 0.074. In the second scenario, adding the time optimization has little effect on flight time. However, the unit time cost of the proposed controller is the lowest compared to the DDPG and PPO controllers.

As analyzed in Section Limitations of existing RL algorithms in the atypical MDPs, the estimated value functions in existing RL algorithm, e.g., DDPG and PPO, is biased due to the TD learning method. Meanwhile, the sampling error $err(s_t)$ can further increase the estimation bias of the state-value function for the stochastic policy algorithms, as analyzed in Section Atypical MDPs: Definition and characteristic analyses. These estimation biases have adverse effects on the policy update. However, due to the average reward method (see Section The immediate-return RL algorithm), an unbiased target Q-value is provided for the proposed algorithm. Thus, the disadvantages of the estimation bias can be overcome. According to the above test data, the effective rate of control Re of the proposed controller in the first scenario is increased by 18.9% than the DDPG controller and increased by 17.7% than the PPO controller. In the second scenario, the effective rate of control Re of the proposed controller is increased by 6.6% than the DDPG controller and increased by 73.6% than the PPO controller. The proposed algorithm also shows better time cost optimization than DDPG and PPO in both

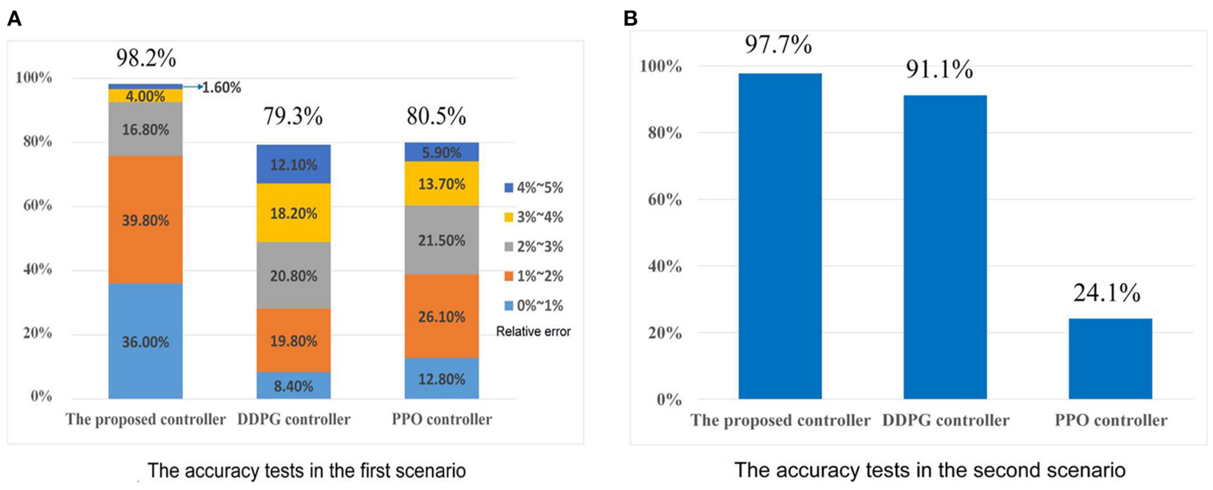


FIGURE 9
The accuracy tests. **(A)** The accuracy test's results in the first scenario. **(B)** The accuracy test's results in the second scenario.

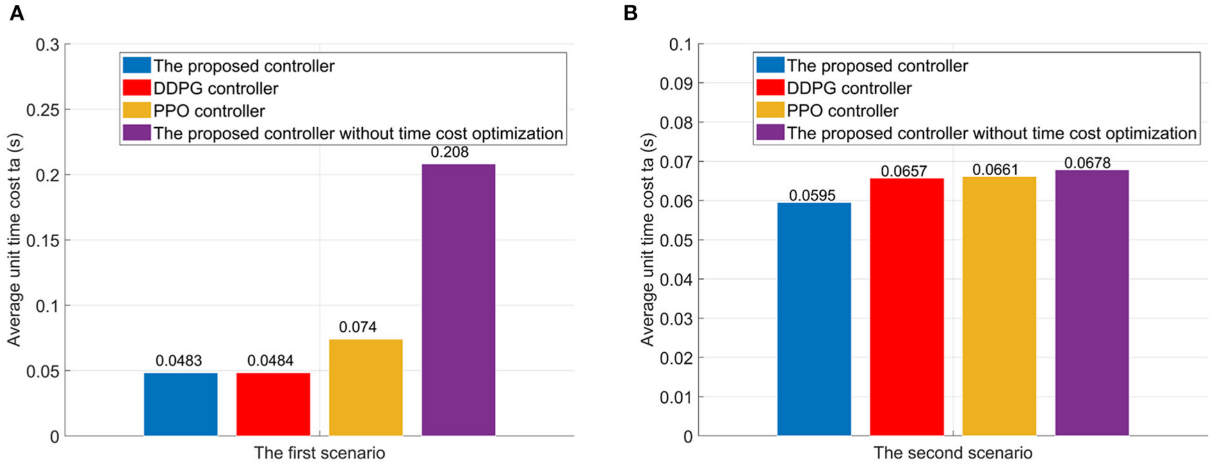


FIGURE 10
The average unit time cost in flights. **(A)** The first scenario. **(B)** The second scenario.

two scenarios. Thus, the high accuracy and low unit time cost of the proposed controllers can be verified. This also means that the immediate-return RL algorithm has better performance than existing RL algorithms in deal with the atypical MDPs.

In the reliability tests, several specified states will be chosen for the tested controllers in each scenario (see Figure 11). For each chosen state, the only set of definite initial velocities will be outputted by the corresponding controller. Then, in the uncertain environment, 200 different flight trajectories will be generated based on the same chosen states and the same initial velocities. To evaluate the reliability of the controllers, the reliable rate Rr is defined as the effective rate of control of the repeated 200 tests on the same chosen state, which is written as Equation (53)

$$Rr = N_{Rr}/200 \tag{53}$$

where N_{Rr} is the number of the flight trajectories controlled successfully in 200 reliability tests.

In the first scenario, a point is selected as the initial position of the moving player. The moving player is assumed to move along the four directions marked by the orange arrows in Figure 11A now. That is, four states are chosen for the tested controllers. According to Figure 12, the average reliable rate of the proposed controller for the first scenario is 100.00%.

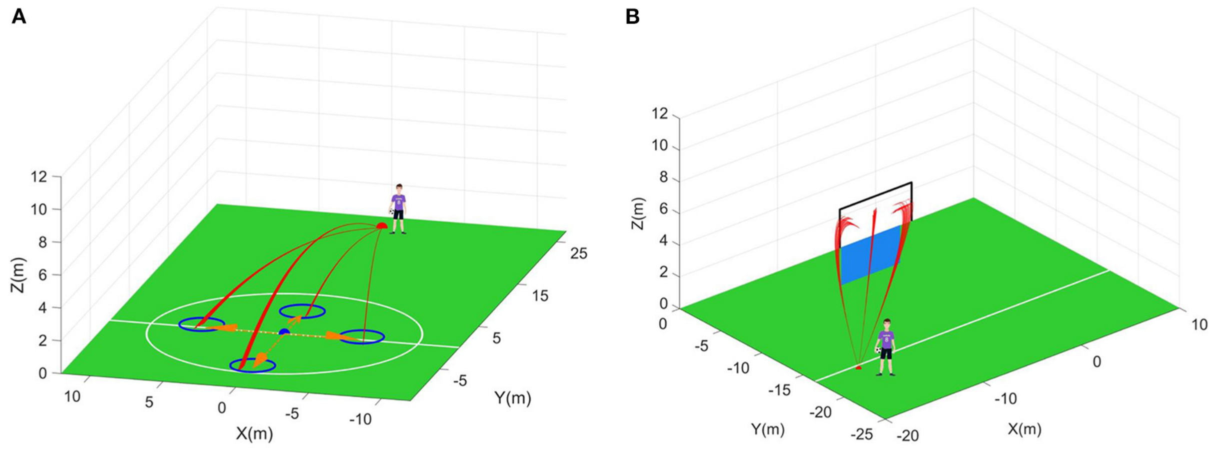


FIGURE 11
Reliability tests. **(A)** The first scenario. Blue circle is the allowed landing range. **(B)** The second scenario. Blue plane is the human wall. Black wireframe is the goal.

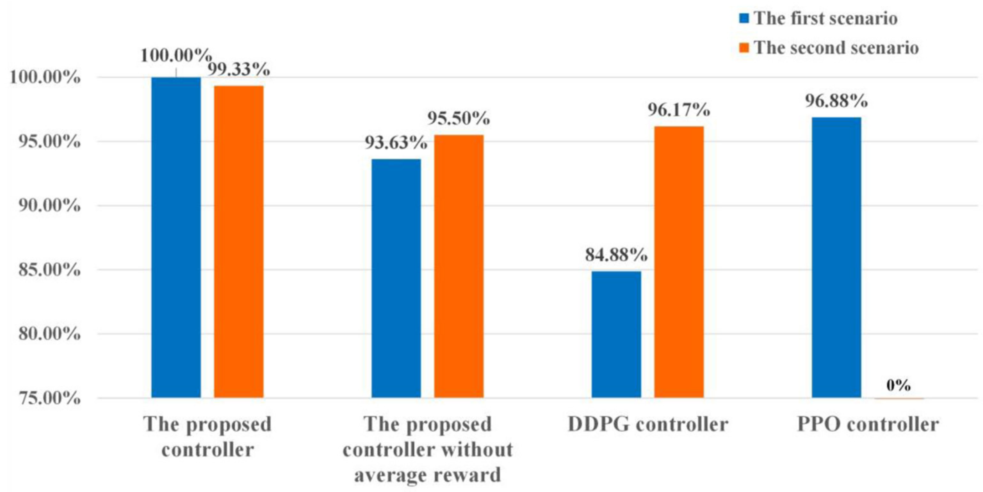


FIGURE 12
The results of the reliability tests.

The average reliable rates of the DDPG controller and PPO controller are 84.88 and 96.88% respectively. In the second scenario, one point is selected as the initial take-off position of the football (Figure 11B). In this initial take-off position, three specified directions where the football flies over the human wall are tested. That is, three states are constructed in the second scenario to test controllers. In this scenario, only 4 trajectories are not control in the total of 600 trajectories under the effect of the proposed controller. The average reliable rate of the proposed controller is 99.33%. The DDPG's average reliable rate in the second scenario is 96.17%. Notice that the PPO controller do not finish the reliability tests due to its terrible control policy.

The reliability in uncertain environments is also an important index to evaluate the controller's performance. In this paper, the aerodynamic model of football with parameter uncertainties is regarded as the uncertain environment. Due to the strong non-linear of the football model, there may be more than one set of initial velocities to meet the requirements of the specified flight purpose. Meanwhile, the same initial velocities may generate different trajectories due to the parameter uncertainties. Thus, high reliability means that the expected reward under the specified state-action pair can be estimated accurately. And the controller can find a good set of initial velocities from multiple possible initial velocities, reducing the effects of the parameter uncertainties on the flight trajectories.

TABLE 5 Computing resources usage tests.

	CPU utilization	Memory utilization (GB)	Computing time (s)	Size of the networks weights (KB)
The proposed algorithm	26%	1.4	2,359	4,682
DDPG	32%	1.9	3,342	6,243
PPO	30%	1.6	2,408	5,455

According to test data in Figure 12, the reliabilities of the proposed controllers are approaching or equal to 100% in both two football flight scenarios, which is significantly better than DDPG and PPO controllers. The above results verify that the proposed controllers have great reliability and can find the best initial velocities to resist the adverse effects of uncertain environments. As analyzed in Section The immediate-return RL algorithm, the great reliability of the proposed controllers come from the average operation for reward. For the sake of comparison, two controllers based on the proposed algorithm without using the average reward are also trained. As shown in Figure 12, the reliable rate of the controller without the average reward is reduced by 6.37% in the first scenario and reduced by 3.83% in the second scenario. Numerical results indicate that the average reward method can improve the reliability of the controller.

Computing resource usage

As analyzed in Section Complexity analysis, compared to existing RL algorithms, the network framework of the immediate-return RL algorithm is greatly simplified, and its complexity is reduced significantly. That is, when solving the same problem in the atypical MDPs, the immediate-return RL algorithm may consume fewer computing resources than existing RL algorithms. Therefore, taking the first scenario of the football trajectory control as an example, the computational resource requirements of different algorithms, i.e., immediate-return RL algorithm, DDPG, and PPO, are analyzed.

In these tests, the hardware is a normal computer with Intel I5 8600k processor and Nvidia GPU RTX2060. And all networks are built by the Tensroflow. For unity, 300,000 training steps are provided for each tested algorithm. Then, the computing resources consumed by three tested algorithms are shown in Table 5. As can be seen, the immediate-return RL algorithm reduces the CPU utilization by 18.8%, the memory utilization by 26.3%, computing time by 29.4%, and size of the networks by 25.0% than the DDPG. Compared to PPO, the immediate-return RL algorithm also reduces the CPU utilization by 13.3%, the memory utilization by 12.5%, computing time by 2.0%, and size of the networks by 14.2%. It should be noticed that the number of training steps is limited to 300,000 in all tests. However, the computing resource usage of the algorithms also depends on the number of training steps required. Since the convergence speed of both DDPG and PPO is slower than the proposed algorithm, they require much more training steps than the proposed algorithm in actuality (see Figure 8). As analyzed in Section Training process, the number of training steps used by the proposed algorithm is 66.7% of the DDPG and 20% of the PPO. That is, the advantage of proposed algorithm in computing time is greater than that shown in the Table 5. Thus, the test data demonstrates that, when dealing with the same problem in the atypical MDPs, the immediate-return RL algorithm trains faster, occupies less CPU and Memory, and generates fewer networks than existing RL algorithms. Furthermore, it should be noted that the transfer processes of data between CPU and GPU also consumes computing resources. The simulations of the football flight also affect the usage of computing resources. Thus, the differences between the comparison results and the theoretical analysis in Section Complexity analysis are acceptable.

Conclusion

The atypical MDPs exist widely in the engineering field, which involves one state transition with continuous action space. The control goal of the atypical MDPs is to maximize the immediate returns. However, the existing RL algorithms are designed for standard MDPs to maximize long-term returns. Thus, they can cause significant estimation errors for the value function and a waste of computing resources when dealing with the atypical MDPs. To solve such problems, this paper analyzes the characteristics of the atypical MDPs systematically and explains the differences between estimating the state-value function and estimating the action-value function. On this basis, the immediate-return RL algorithm was proposed to deal with the atypical MDPs. In the proposed algorithm, the method of average reward is developed to provide the unbiased and low variance target Q-value. Thus, the problems of large estimation errors can be overcome. And a newly designed network framework is designed for the proposed algorithm, which can significantly reduce computing resource usage. Then, two scenarios of the football trajectory control, i.e., passing the football to a moving player, and chipping the football over the human wall, are designed as the benchmark to test the algorithms designed for the atypical MDPs. Numerical results demonstrate that the learning efficiency of the proposed algorithm is 1.5 times that of the DDPG and 5 times that of the PPO. For the controllers based on the proposed algorithm, their effective rates of control are more than 97.7%, and their reliabilities are approaching 100%. Such performance is far superior to DDPG and PPO. As the proposed controller increases the shot's accuracy significantly, it can promote the development of high-level football robots in the Robot world cup. Furthermore, the proposed algorithm can also consume fewer computing resources than existing RL algorithms. Thus, the immediate-return RL algorithm has higher learning efficiency, higher performance, and lower computing resource usage than the existing RL algorithms, such as PPO and DDPG.

It should be pointed out that the immediate-return RL algorithm can output only one determined action. This determined value can be seen as the best solution according to the specified rewards function. However, a single best solution based on the specified rewards function is impractical for many complex engineering problems (e.g., strongly non-linear dynamic system with parameter uncertainties). As one focus of the future work, efforts will be made to improve the algorithm to find a proper basin which corresponds to the specified scenario. After that, the action output shall be more practical. In the future, we will devote ourselves to expand the use of the proposed immediate-return RL algorithm and achieve more engineering applications, such as stamping process, directional blasting, approximations of the compound Poincaré maps, etc.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

ZP contributed to algorithm design and development, data analysis, and writing the first manuscript. GW contributed to the study conception and design and supervised the overall study. ZT assisted in implementing the code and collecting datasets. SY guided the research and provided a critical review. XH collected related literature and assisted in the data processing. All authors reviewed and approved the final manuscript.

Funding

This work was supported by National Natural Science Foundation of China (No. 11832009) and Science and technology innovation Program of Hunan Province (No. 2020RC2027).

Acknowledgments

We are very grateful to the reviewers for their valuable comments and suggestions.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bellman, R. (1957). A Markovian decision process. *J. Mathem. Mech.* 6, 679–684. doi: 10.1512/iumj.1957.6.56038

Brys, T., Harutyunyan, A., Vrancx, P., Nowé, A., and Taylor, M. E. (2017). Multi-objectivization and ensembles of shapings in reinforcement learning. *Neurocomputing.* 263, 48–59. doi: 10.1016/j.neucom.2017.02.096

Chen, L., Jiang, Z., Cheng, L., Knoll, A. C., and Zhou, M. (2022). Deep reinforcement learning based trajectory planning under uncertain constraints. *Front. Neurorob.* 16, 883562. doi: 10.3389/fnbot.2022.883562

Dewey, D. (2014). "Reinforcement learning and the reward engineering principle," in *2014 AAAI Spring Symposium Series.*

Djaoui, L., Chamari, K., Owen, A. L., and Dellal, A. (2017). Maximal sprinting speed of elite soccer players during training and matches. *J. Strength Condit. Res.* 31, 1509–1517. doi: 10.1519/JSC.0000000000001642

Fujimoto, S., Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning,* 1587–1596. PMLR.

Han, S., Pool, J., Tran, J., and Dally, W. (2015). "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems,* 28.

He, K., and Sun, J. (2015). "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 5353–5360. doi: 10.1109/CVPR.2015.7299173

Henderson, P., Islam, R., Bachman, P., Pineau, J., and Precup, D. (2018). "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*. doi: 10.1609/aaai.v32i1.11694

Horowitz, M., and Williamson, C. (2010). The effect of Reynolds number on the dynamics and wakes of freely rising and falling spheres. *J. Fluid Mech.* 651, 251–294. doi: 10.1017/S0022112009993934

Hou, Z., Chi, R., and Gao, H. (2016). An overview of dynamic-linearization-based data-driven control and applications. *IEEE T Ind. Electron*. 64, 4076–4090. doi: 10.1109/TIE.2016.2636126

Hou, Z., and Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Inform Sci*. 235, 3–35. doi: 10.1016/j.ins.2012.07.014

Javorova, J., and Ivanov, A. (2018). "Study of soccer ball flight trajectory," in *MATEC Web of Conferences*. EDP Sciences, 01002. doi: 10.1051/matecconf/201814501002

Kiratidis, A. L., and Leinweber, D. B. (2018). An aerodynamic analysis of recent FIFA world cup balls. *Eur. J. Phys*. 39, 34001. doi: 10.1088/1361-6404/aaa888

Lee, J., Koh, H., and Choe, H. J. (2021). Learning to trade in financial time series using high-frequency through wavelet transformation and deep reinforcement learning. *Appl. Intell.* 51, 6202–6223. doi: 10.1007/s10489-021-02218-4

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.

Li, Z., Qiao, L., Jiang, J., Hong, L., and Sun, J. (2020). Global dynamic analysis of the North Pacific Ocean by data-driven generalized cell mapping method. *Int. J. Dynam. Control*. 8, 1141–1146. doi: 10.1007/s40435-020-00678-z

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Massi, E., Barthélemy, J., Mailly, J., Dromnelle, R., Canitrot, J., Poniatowski, E., et al. (2022). Model-Based and Model-Free Replay Mechanisms for Reinforcement Learning in Neurorobotics. *Front. Neurorobot*. 16, 864380. doi: 10.3389/fnbot.2022.864380

Minsky, M. L. (1954). *Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem*. Princeton, NJ: Princeton University.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016). "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning,* 1928–1937). PMLR.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature.* 518, 529–533. doi: 10.1038/nature14236

Myers, T. G., and Mitchell, S. L. (2013). A mathematical analysis of the motion of an in-flight soccer ball. *Sports Eng.* 16, 29–41. doi: 10.1007/s12283-012-0105-8

Neilson, P. J. (2003). *The Dynamic Testing of Soccer Balls.* Loughborough, UK: Loughborough University.

Norman, A. K., and McKeon, B. J. (2011). Unsteady force measurements in sphere flow from subcritical to supercritical Reynolds numbers. *Exp. Fluids.* 51, 1439–1453. doi: 10.1007/s00348-011-1161-8

Pan, Z., Yin, S., Wen, G., and Tan, Z. (2023). Reinforcement learning control for a three-link biped robot with energy-efficient periodic gaits. *Acta Mechan. Sinica*. 39, 522304. doi: 10.1007/s10409-022-22304-x

Schulman, J. (2016). *Optimizing expectations: From deep reinforcement learning to stochastic computation graphs*. UC Berkeley.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Sharbafi, M. A., Azidehak, A., Hoshyari, M., Bakhshandeh, O., Babarsad, A. A. M., Zareian, A., et al. (2011). "MRL extended team description 2011," in *Proceedings of the 15th international RoboCup symposium, Istanbul, Turkey* (pp. 1-29).

Silver, D., Singh, S., Precup, D., and Sutton, R. S. (2021). Reward is enough. *Artif. Intell.* 299, 103535. doi: 10.1016/j.artint.2021.103535

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* Cambridge, MA: The MIT Press.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. (1999). "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems,* 12.

Tutsoy, O., and Brown, M. (2016). Chaotic dynamics and convergence analysis of temporal difference algorithms with bang-bang control. *Optimal Control Applic. Methods.* 37, 108–126. doi: 10.1002/oca.2156

Van Hasselt, H., Guez, A., and Silver, D. (2016). "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*. doi: 10.1609/aaai.v30i1.10295

Wang, N., and Budiansky, B. (1978). Analysis of sheet metal stamping by a finite-element method. *J. Appl. Mech.* 45, 73–82. doi: 10.1115/1.3424276

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). "Dueling network architectures for deep reinforcement learning," in *International Conference on Machine Learning,* 1995–2003. PMLR.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, University of Cambridge.

Zhu, Z., Xie, H., and Mohanty, B. (2008). Numerical investigation of blasting-induced damage in cylindrical rocks. *Int. J. Rock. Mech. Min.* 45, 111–121. doi: 10.1016/j.ijrmms.2007.04.012

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 15 December 2022
DOI 10.3389/fnbot.2022.1091361

Check for updates

OPEN ACCESS

EDITED BY
Chen Qiao,
Xi'an Jiaotong University, China

REVIEWED BY
Aite Zhao,
Qingdao University, China
Fuqiang Gu,
Chongqing University, China

*CORRESPONDENCE
Ziliang Ren
renzl@dgut.edu.cn

RECEIVED 07 November 2022
ACCEPTED 28 November 2022
PUBLISHED 15 December 2022

CITATION
Yang H, Ren Z, Yuan H, Wei W, Zhang Q and Zhang Z (2022) Multi-scale and attention enhanced graph convolution network for skeleton-based violence action recognition.
Front. Neurorobot. 16:1091361.
doi: 10.3389/fnbot.2022.1091361

Multi-scale and attention enhanced graph convolution network for skeleton-based violence action recognition

Huaigang Yang[1], Ziliang Ren[1]*, Huaqiang Yuan[1], Wenhong Wei[1], Qieshi Zhang[2] and Zhaolong Zhang[3]

[1]School of Computer Science and Technology, Dongguan University of Technology, Dongguan, China, [2]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, [3]Rutgers, The State University of New Jersey, New Brunswick, NJ, United States

Graph convolution networks (GCNs) have been widely used in the field of skeleton-based human action recognition. However, it is still difficult to improve recognition performance and reduce parameter complexity. In this paper, a novel multi-scale attention spatiotemporal GCN (MSA-STGCN) is proposed for human violence action recognition by learning spatiotemporal features from four different skeleton modality variants. Firstly, the original joint data are preprocessed to obtain joint position, bone vector, joint motion and bone motion datas as inputs of recognition framework. Then, a spatial multi-scale graph convolution network based on the attention mechanism is constructed to obtain the spatial features from joint nodes, while a temporal graph convolution network in the form of hybrid dilation convolution is designed to enlarge the receptive field of the feature map and capture multi-scale context information. Finally, the specific relationship in the different skeleton data is explored by fusing the information of multi-stream related to human joints and bones. To evaluate the performance of the proposed MSA-STGCN, a skeleton violence action dataset: Filtered NTU RGB+D was constructed based on NTU RGB+D120. We conducted experiments on constructed Filtered NTU RGB+D and Kinetics Skeleton 400 datasets to verify the performance of the proposed recognition framework. The proposed method achieves an accuracy of 95.3% on the Filtered NTU RGB+D with the parameters 1.21M, and an accuracy of 36.2% (Top-1) and 58.5% (Top-5) on the Kinetics Skeleton 400, respectively. The experimental results on these two skeleton datasets show that the proposed recognition framework can effectively recognize violence actions without adding parameters.

KEYWORDS

violence action recognition, skeleton sequence, multi-scale graph convolution network, attention mechanism, spatiotemporal information

1. Introduction

Recently, individual and group violence in public places has seriously threatened the safety of people's lives and property. With the widespread deployment of video surveillance equipment, video motion understanding and recognition based on computer vision technology has become an effective public security tool for identifying danger and preventing crime. However, the detected targets in surveillance scenes are often affected by background noise, light intensity changes, camera views, and clothing, which requires not only improving the accuracy of the model but also considering the computational cost of the algorithm (Serrano et al., 2018; Ramzan et al., 2019). The existing recognition methods mainly use different modalities as inputs, and learn spatiotemporal features by designing Convolutional Neural Networks (CNN) (Cheng et al., 2021; Ji et al., 2021; Gadelkarim et al., 2022) and Recurrent Neural Networks (RNN) (Liu et al., 2018; Song et al., 2018; Jiang et al., 2020; Shu et al., 2021).

With the development of a graph convolution network (GCN), the skeleton-based approaches have achieved success in violent action recognition due to it can better reduce the model complexity (Senst et al., 2017; Liu Z. et al., 2020; Li M. et al., 2022). The skeleton data is essentially a topological graph, where human joints are represented as vertices and bones are represented as edges of the graph. Although skeleton sequences has comparative advantages over RGB or depth modalities, skeleton based recognition methods still face difficulties and challenges in the following two aspects: (1) In the spatial space, there is spatial information and a certain strong correlation between the neighboring nodes in each frame, and it is necessary to mine the action structure information. (2) In the temporal space, the motion structure of the joint points is important for characterizing the action, which needs to model the long-range temporal information.

As existing work mainly considers a series of convolution operations on a single feature map (Liu Z. et al., 2020; Li M. et al., 2022), which to some extent fails to obtain larger receptive field information. We use the design of a multi-scale approach to obtain larger and more receptive field information, which is beneficial for feature learning of the model and expression. The attention mechanism mainly focuses the model on the main joint points or skeletal edges where certain movements occur, which helps to eliminate redundant dependency information between joint point features, thus effectively capturing the main association information between joint points. Meanwhile, thanks to advanced pose estimation methods (Openpose, Cao et al., 2021) the skeleton information may be extracted from the RGB video easily and efficiently. To improve the recognition accuracy and reduce the computational cost, this paper proposes a multi-scale GCN with data preprocessing and attention modules to extract spatiotemporal information and combine multi-stream features for skeleton-based violent action recognition. Firstly, the spatial GCN with the attention module is constructed to extract the multi-scale spatial features by learning the adjacency information between the multi-order joints and build the channel-based dependencies with a low number of parameters. And then, a temporal GCN in the form of hybrid dilation convolution to obtain different sizes of perceptual fields and extract the multiscale contextual information by setting different dilation convolution rates. Finally, the accuracy of recognition is further improved by fusing the multi-stream features related to human joints and bones.

The main contributions of this paper are as follows:

(1) In the spatial space, we design a multi-scale spatial GCN with a fused channel attention mechanism to extract spatial information and the correlation features between channels.
(2) In the temporal space, we propose a temporal convolution network in the form of hybrid dilation convolution to extract the temporal features from skeleton sequences, which can be used to capture multi-scale contextual information and reduce the number of network parameters.
(3) The model incorporates joint position, joint motion, bone vector and bone motion information to further improve the accuracy of violent action recognition.

2. Related works

In the field of computer vision, deep learning approaches have become the dominant research direction in tasks such as image classification and target detection since they have a better ability to capture distinguishing features. In this paper, three categories of deep learning methods based on skeleton sequences are briefly reviewed: CNN, RNN, and GCN.

2.1. CNN-based methods

Since CNNs can learn high-level semantic information efficiently and effectively, they are usually widely used in image processing tasks. However, it is difficult and challenging to balance and make full use of spatiotemporal information for human action recognition based on skeleton sequences (Kim and Reiter, 2017). The mainstream approaches usually represent skeleton sequences as pseudo images as the standard input of CNNs (Cao et al., 2018; Hou et al., 2018; Xu et al., 2018; Li C. et al., 2019). In these methods, the spatial structure and temporal dynamic information of the skeleton sequences are encoded as columns and rows of a tensor, respectively. Caetano et al. (2019b) proposed a method to represent skeletal motion information based on convolution neural networks, which first encoded the temporal dynamic information by calculating the magnitude and direction values of the joint motion, and then different time scales were used to filter the noisy motion information for capturing long-distance joint point dependence. In addition, Caetano et al. (2019a) introduced reference nodes and tree structures to represent the skeleton image through the framework of the SkeleMotion method, the former incorporating different spatial information among the articulations, but the latter preserving important spatial relationships by traversing a skeleton tree with a depth-first algorithm. By considering only adjacent joints within the convolution kernel to learn co-occurring features, some potentially associated joints are ignored. Therefore, Li C. et al. (2018) used an end-to-end network framework to learn co-occurrence features by a hierarchical approach in which contextual information is gradually aggregated at different layers. First, point-level information is encoded independently for each node. Then, combining them into semantic representations in the temporal and spatial domains, respectively.

2.2. RNN-based methods

The RNN-based approaches essentially uses the output of the previous frame as the input of the current frame, which allows continuous sequential data to be processed efficiently. To remedy the gradient disappearance and long-range modeling problems of standard RNN, researchers have proposed improved RNNs such as long short-term memory neural network (LSTM) and gated neural unit (GRU), which model the spatiotemporal dimension to capture the correlation features between sequence data (Liu et al., 2018; Song et al., 2018; Jiang et al., 2020; Shu et al., 2021). Wang and Wang (2017) proposed a two-stream recurrent neural network to model spatiotemporal information by using 3D transforms-based data enhancement techniques. To extract more distinguished spatiotemporal features, Song et al. (2017) proposed two spatiotemporal attention sub-modules based on LSTM networks and designed a spatial attention sub-module based on the joint selection gate, which can adaptively assign attention weights to the skeleton nodes in each frame. Meanwhile, the temporal attention sub-module based on the frame selection gate is designed to assign different attention weights to different frames for the extraction of keyframes. A longer and deeper RNN network is proposed by Li S. et al. (2018) to solve the gradient explosion and disappearance problem, which be constructed to learn high-level semantic features with better robustness. Furthermore, due to the strong capability of CNNs for spatial feature extraction, Li C. et al. (2022) combined RNN and CNN models to improves the spatiotemporal modeling capability in complex scenes, as RNN is mainly used for temporal modeling and CNN is mainly used for spatial modeling.

2.3. GCN-based methods

The human skeleton sequence is inherently a topological graph, rather than a Euclidean spatial image based on CNNs or a segment of sequence vectors based on RNNs methods. The spatiotemporal dependencies between the associated vertices cannot be fully expressed by simply transforming the sequence into a two-dimensional pseudo-image or sequence vector. The GCN is developed based on CNN (Gao et al., 2019; Si et al., 2019; Wu et al., 2019; Degardin et al., 2021; Tu et al., 2022), which can be used to efficiently capture spatial features information by adjusting the convolution kernel size with different neighbors of each vertex. Yan et al. (2018) proposed a spatiotemporal graph convolutional neural network (ST-GCN) for human behavior recognition, which consider human joints as vertices of a graph and connections between joints and different frames of the same joints as edges of the graph. By designing different convolutional kernel strategies for modeling, the spatiotemporal features between joints are captured and the action is predicted by a Softmax classifier. As the skeleton graph used in ST-GCN, there is an implicit problem of missing node-dependence. To obtain richer inter-joint dependencies, Li M. et al. (2019) proposed an action-structural graph convolutional neural network (AS-GCN) with an actional-links module to extend the skeleton graph to represent higher-order dependencies and capture the potential dependencies of a specific action. Shi et al. (2019b) proposed a two-stream adaptive graph convolution network (2s-AGCN) for adaptive learning of spatiotemporal features from skeleton sequences in end-to-end networks. Similarly, Li B. et al. (2019) proposed a spatiotemporal graph routing (ST-GR) approach to capture the intrinsic higher-order connectivity relationships among the skeleton joints, which added additional edges to the network skeleton graph through a global self-attentive mechanism. Liu Z. et al. (2020) proposed a decomposed multiscale aggregation method and a spatiotemporal graph convolution operator (G3D) to implement a powerful feature extractor. Zhang et al. (2020) proposed a simple effective semantics-guided neural network (SGN) to obtain higher-order semantic information of the nodes for skeleton-based action recognition. To reduce the computational cost of the GCN, Cheng et al. (2020) designed a Shift-GCN that employs a shift-graph operation and a point-level convolution form instead of using standard graph convolution. Along this line of research, Song et al. (2022) proposed a multi-stream GCN model that incorporates input branches including joint position, motion velocity and skeletal features at an early stage, and utilizes separable convolutional layers and a composite scaling strategy to reduce significantly redundant trainable parameters while increasing model capacity. Recently, Chen et al. (2021) proposed a channel-level topology refinement graph convolution (CTR-GC) based on dynamic topology and multi-channel feature modeling. Specifically, CTR-GC takes the shared topology matrix as the entire prior for a channel and then refines it by inferring channel-specific correlations to obtain a channel-level topology. Li et al. (2021) proposed an Elastic Semantic Network (Else-Net), which consists of a GCN backbone model and multiple layers of elastic units for continuous human behavior recognition. In particular, each flexible unit contains several learning blocks to learn diverse knowledge from different human behaviors, with a switch block to select the most relevant block for the newly entered behavior. Chi et al. (2022) proposed InfoGCN that includes an information bottleneck goal to learn maximally informative action representations and an attention-based graph convolution to infer contextually relevant skeletal topology.

3. Proposed method

3.1. Overall framework

Inspired by the success of the two-stream framework and graph convolution (Shi et al., 2019b, 2020), this paper proposes a multi-scale attention spatiotemporal graph convolution network (MSA-STGCN) to recognize violence human actions from different perspectives, as shown in Figure 1. First, the original joint data are preprocessed to obtain joint position, bone vector, joint motion and bone motion information. Then, these four categories of data are input into our designed MSA-STGCN, respectively. Finally, the four-stream features are fused using a weighted summation method to predict the action category.

3.2. The proposed MSA-STGCN

The proposed MSA-STGCN consists of nine spatiotemporal feature extraction modules, as shown in Figure 2. Given a skeleton sequence $X \in \mathbb{R}^{C\times T\times V}$, where C, T, and V are the number of channels, sequences and joint points of the input data, respectively. Among them, the batch normalization layer (BN) normalizes the input data X, the output feature size of modules B$_1$ to B$_3$, B$_4$ to B$_6$, and B$_7$ to B$_9$ are $B \times C \times T \times V$, $B \times C \times T/2 \times V$, and $B \times C \times T/4 \times V$, respectively, where B is the number of batch size, and the number of output channels of modules are 96, 96, 96, 192, 192, 192, 384, 384, and 384, respectively. Modules B$_1$, B$_4$, and B$_7$ adopt the multi-scale attention enhanced spatial graph convolution network (MSA-SGCN) to extract the spatial features, while modules B$_2$, B$_3$, B$_5$, B$_6$, B$_8$, and B$_9$ use multi-scale temporal graph convolution networks (MS-TGCN) to obtain the temporal feature from skeleton sequences. Then, global average pooling (GAP) layer is applied to aggregate the spatiotemporal features and unify the feature graph size of the samples. Finally, the Softmax layer is used to obtain the classification probability and prediction category.

3.2.1. Multi-scale attention enhanced spatial graph convolution network

The effectiveness of the attention mechanism has been demonstrated in tasks such as target detection and image classification, which has been gradually introduced into the field of action recognition. In this paper, we design a channel attention module based on the Squeeze-and-Excitation Networks (SE-Net) (Hu et al., 2020), named multi-scale attention Spatial Graph Convolution Network (MSA-SGCN), to automatically learn the correlation and significance between feature map channels. The SE-Net improves the feature description capability by modeling the dependencies of each channel, which enhances useful features and suppress non-useful features by adaptively adjusting the feature response values of each channel. Motivated by these advantages, we insert the Squeeze-and-Excitation module to a spatial graph of the convolution neural network to obtain more contextual feature through automatically learning the importance of different channel features. The earliest application of GCNs to human action recognition tasks is ST-GCN, where spatiotemporal graph convolution and spatial division strategies are used to model skeleton sequences to extract feature information in the spatial space (Yan et al., 2018). In contrast, a multi-scale spatial and motion graph convolution modules are designed in STI-GCN (Huang et al., 2020) to extract and merge features for topological graphs from multiple perspectives.

Based on the success of these models, we design a multi-scale attention spatial graph convolution network to learn spatial features from skeleton sequences, as shown in Figure 3. The feature extraction for each input layer is performed by

$$X_t^{l+1} = ReLU(\sum_k D_k^{-\frac{1}{2}} A_k D_k^{-\frac{1}{2}} X_t^l W_k^l) \tag{1}$$

where k controls the scale size of the whole network and also represents the shortest distance between the nodes V_i and V_j. A_k represents the relationship matrix between the current node and the k-hop neighbors, which includes the self-loop connections. It allows the model to learn information about the neighbor's features between each node. D_k denotes the square root of the inverse of the degree matrix of the neighborhood matrix A_k, which is used for symmetric normalization of the neighborhood matrix A_k. In the calculation, the features of the node itself have been calculated as well as the weighted sum of the features of all neighbors. X_t represents the input of the frame and denotes the number of layers of the network. W_k is the current node, W_k is a learnable weight matrix between

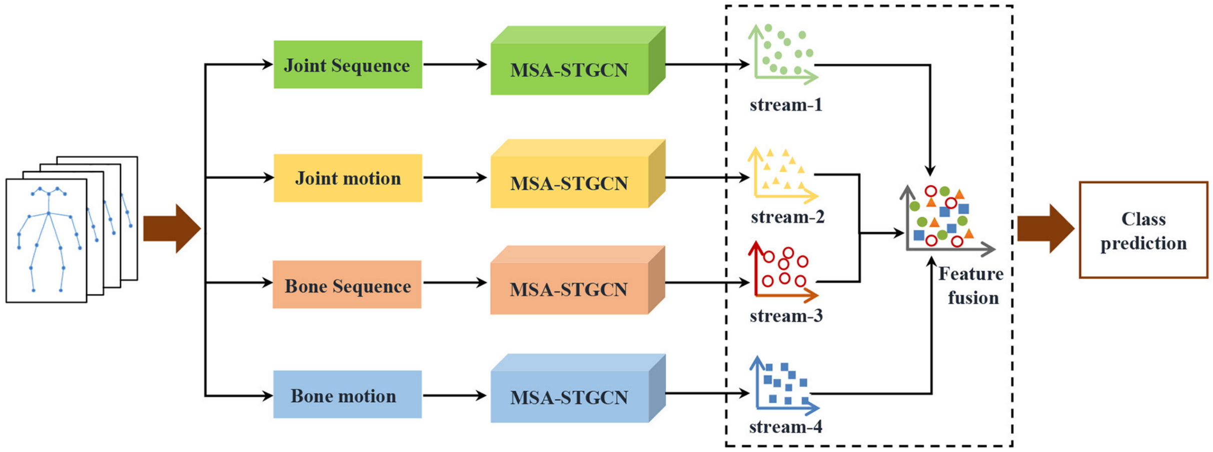


FIGURE 1
Multi-stream fusion violence action recognition framework.

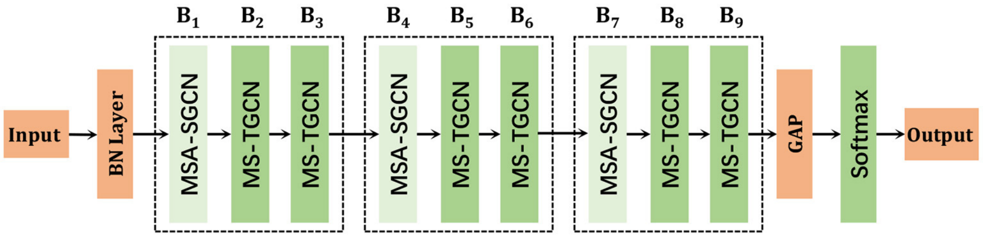


FIGURE 2
Multi-scale attention spatiotemporal graph convolution network.

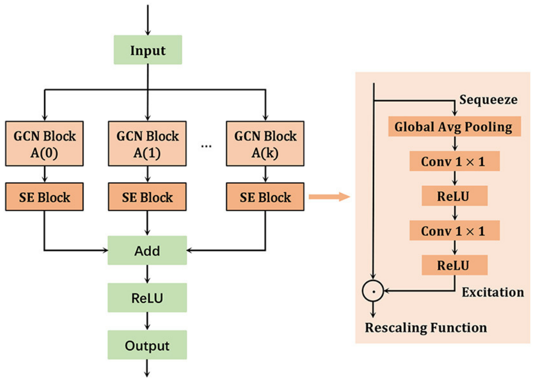


FIGURE 3
Multi-scale attention enhanced spatial graph convolution network.

the *k*-hop neighbors of the current node, which implements the edge importance weighting. *Relu*() represents the activation function.

In the proposed MSA-SGCN, the scale of each model is adjusted by *k* to obtain different scale feature information in the spatial space, and the multi-order neighborhood information is aggregated to obtain all the neighborhood feature information of each joint. In addition, attention operations are performed on each scale output feature in the channel dimension to automatically learn the correlation contextual information between feature map channels.

3.2.2. Multi-scale temporal graph convolution network

Existing methods usually use standard convolution with fixed kernel size throughout the network module to model the temporal information (Yan et al., 2018; Li M. et al., 2019; Shi et al., 2019a,b). In this paper, we proposed a multiscale aggregation learning method by introducing hybrid dilation convolution to improve the traditional temporal convolution module (Ople et al., 2020). Because of the exponential expansion of the perceptual field with guaranteed coverage, the proposed MS-TGCN can effectively aggregate multi-scale contextual information without loss of resolution by using dilation convolution. However, the result of a certain layer of null convolution is not dependent on the information of the previous layer due to the grid effect problem of the dilation convolution, and the information obtained from the long-distance convolution lacks relevance. Therefore, this model avoids the grid effect problem by introducing a hybrid form of dilation convolution (Wang et al., 2018). At the same time, the model takes the feature map X as input without introducing additional parameters and generates a feature map of the same size with the same dimension, which is passed to the next network module.

As shown in Figure 4, the number of model parameters is reduced by adopting a multi-branch structure and passing each branch through a convolution kernel of size 1 × 1. The size of the convolution kernels in each branch is modified to 5 × 1, which gives a larger perceptual field than the convolution kernel size of 3 × 1. In addition, we also set the convolution rate of different sizes of holes, 1, 2, and 3 to obtain different scales of the same feature map for avoiding the problem of gradient disappearance. Finally, the aggregation layer fuses the multi-scale information and passes it to the next module of the network. The proposed model can learn richer temporal features and reduce the number of parameters after replacing the regular temporal convolution method.

3.3. Representation of skeleton sequences

The position of the joint points of the human skeleton is defined as:

$$V_{i,t} = \left(x_{i,t}, y_{i,t}, z_{i,t}\right), \forall i \in N, t \in T \qquad (2)$$

where N is the number of joints in the human skeleton, T is the total number of sequences, and i represents the joints at time t. In 3D skeleton sequences, the joint positions consist of three position coordinates (x, y, z), which are usually captured directly by a depth camera or extracted from RGB video data.

$$B_{i,j,t} = V_{j,t} - V_{i,t} = (x_{j,t} - x_{i,t}, y_{j,t} - y_{i,t}, z_{j,t} - z_{i,t}) \qquad (3)$$

In particular, the joint near the center of gravity of the human skeleton is defined as the source node with coordinates denoted as $V_{i,t}$, while the joint far from the center of gravity is defined as the target node with coordinates denoted as $V_{j,t}$. Since each joint has no self-loop, each bone can be assigned a unique joint point, forming a directed acyclic graph. In addition, since the root node does not have any bones assigned to it, to simplify the network design, the vector of bones assigned to the root node is set to 0.

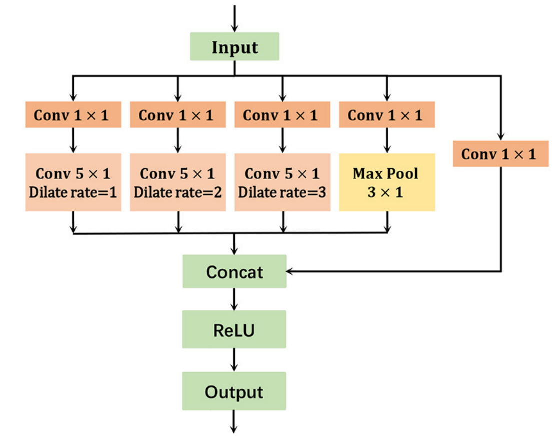


FIGURE 4
Multi-scale temporal graph convolution network.

The definition of human joint motion information is defined as:

$$J{-}M_{i,t+1} = V_{i,t+1} - V_{i,t} = (x_{i,t+1} - x_{i,t}, y_{i,t+1} - y_{i,t}, z_{i,t+1} - z_{i,t}) \qquad (4)$$

where $V_{i,t}$ represents the position coordinates of the i^{th} joint at time $t:(x_{i,t}, y_{i,t}, z_{i,t})$, and $V_{i,t+1}$ represents the position coordinates of the i^{th} joint at time $t+1:(x_{i,t+1}, y_{i,t+1}, z_{i,t+1})$, and the position of the same joint in adjacent frames are difference to obtain the sequence of joint motion information.

The definition of human bone motion information is defined as:

$$B{-}M_{i,j,t,t+1} = B_{i,j,t+1} - B_{i,j,t} \qquad (5)$$

where $B_{i,j,t}$ represents the skeletal vector information at time t, and $B_{i,j,t+1}$ represents the skeletal vector information at time $t+1$. We capture the skeletal motion information by the difference of adjacent skeletal vectors. The fusion strategy is used to gather the features of nodal position information, skeletal vector information, nodal motion information, and skeleton motion information streams.

3.4. Implementation details

The configuration information of the experimental platform is Intel Xeon Silver 4210R CPU with 2.40GHz, 80G memory, 1TB SSD storage, and RTX3090. The number of samples per training batch (Batch size) is set to 32, and the cross-entropy function is used as the loss function for gradient back propagation. The number of iterations (Epoch) is set to 80, and the weight decay parameter is set to 0.0005.The initial learning rate is set to 0.05, and the learning rate is adjusted at a given

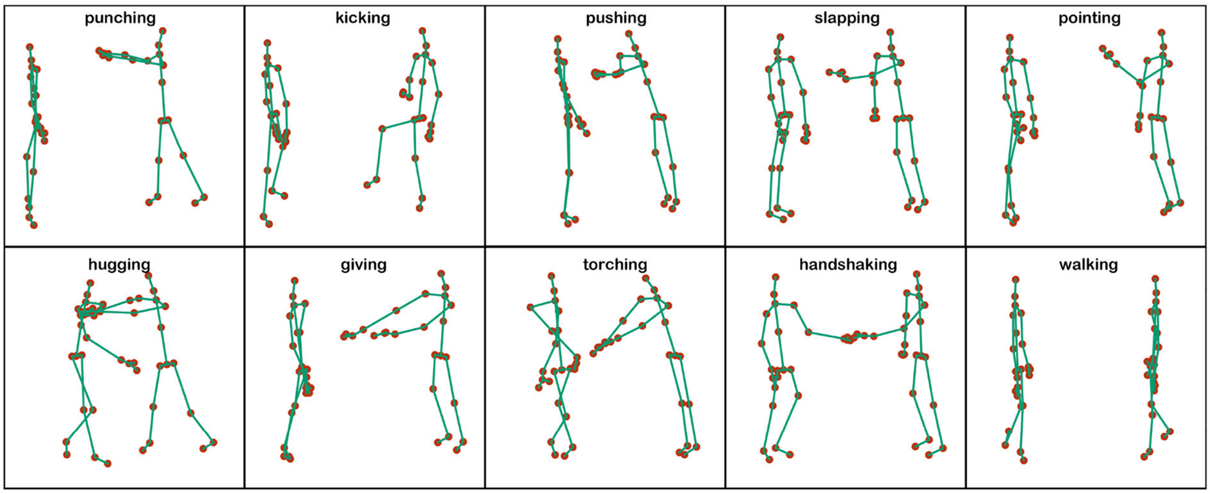


FIGURE 5
Visual representation of 10 types of human violence actions.

interval by dividing the learning rate by 10 when the 30*th* Epoch and 40*th* Epoch are reached, respectively.

4. Experiments

4.1. Datasets

In this paper, we conducted experiments on two datasets: Filtered NTU RGB+D and Kinetics Skeleton 400. The Filtered NTU RGB+D dataset is based on the NTU RGB+D 120 dataset (Liu J. et al., 2020) by discarding other daily movements and filtering out 10 types of movements to form a skeleton dataset. The Kinetics Skeleton 400 dataset is based on the Kinetics-400 dataset (Carreira and Zisserman, 2017) by preprocessing each frame of the original RGB video with a pose estimation algorithm to extract the skeleton sequence data to form a 400 classes normal motion dataset.

4.1.1. Filtered NTU RGB+D

The NTU RGB+D 120 is the largest and most widely used indoor motion dataset, containing 114,400 motion clips in 120 categories. Each clip was performed by 40 volunteers ranging in age from 10 to 35 years old, and each action was filmed from different angles using three Kinect V2 cameras. The previous violence dataset is mainly RGB, depth information, and optical flow modality, while NTU RGB+D 120 is 3D skeleton data, which contains 3-dimensional coordinates of 25 body joints in each frame. Meanwhile, to compare the traditional graphical neural network in a violence recognition task, this paper takes 120 classes of NTU RGB+D 120 dataset for filtering, and finally selected 10 classes of skeleton data about human interaction actions, and the final action types are visualized as shown in Figure 5, including walking, pushing, punching, pointing, slapping, shaking hands, touching, hugging, giving and kicking, among which pushing, punching, kicking, pointing and slapping are the five kinds of video the common violent actions in surveillance. In this paper, we mainly study the recognition of violent actions in surveillance video, and the application scenario is usually the recognition of actions from a certain viewpoint for different objects. Therefore, we adopt a Cross-subject (X-Sub) protocol from the recommended benchmark of the original paper and reports the Top-1 accuracy in the experiment.

4.1.2. Kinetics Skeleton 400

Kinetics-400 is a large human action dataset with 300,000 video clips from the YouTube video site. It covers 400 human action categories from daily life, sports scenes, and complex human interactions. However, this dataset only provides raw RGB video clips without skeleton data. In this work, since the concentration is on skeleton-based action recognition, so we use the OpenPose pose estimation method for preprocessing to extract the coordinates of human joint positions for each frame of each clip. For a multi-person action scene, the two persons with the highest average nodal confidence are selected. In this way, an RGB segment with T-frames is converted into a skeleton sequences. The final dataset consists of a training set of 240,000 segments and a validation set of 20,000 segments. In this paper, we compare the models on the training set and report the accuracy of the validation set. Referring to the evaluation methods proposed in Yan et al. (2018) and Liu Z. et al. (2020),

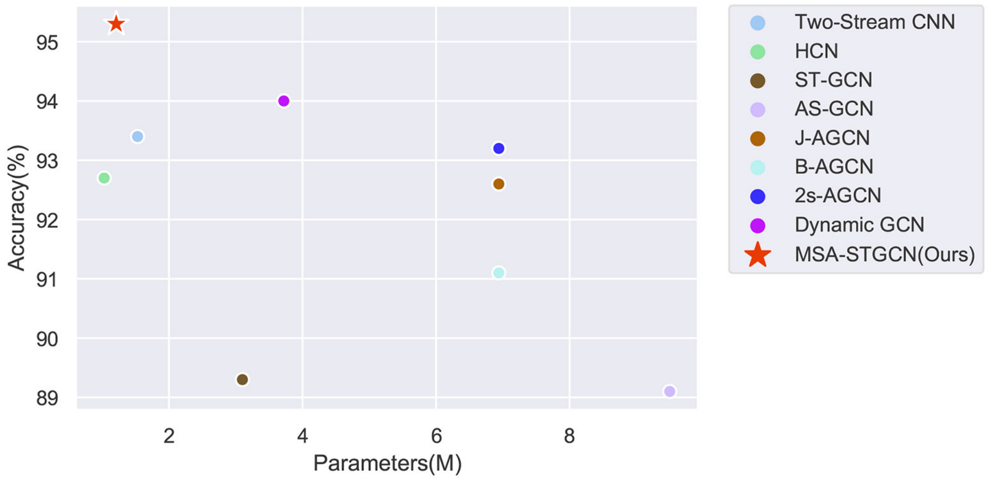


FIGURE 6
The accuracy and parameters of the proposed method compared to other methods on the Filter NTU RGB+D dataset.

we trains the model on the training set and reports the accuracy of Top-1 and Top-5 on the validation set.

4.2. Effectiveness of the proposed method

On the Filtered NTU RGB+D dataset, we have done comparison experiments on two CNN-based methods, namely Two-Stream CNN and HCN model, and on four GCN based methods, namely ST-GCN, AS-GCN, 2S-AGCN and Dynamic GCN network, and the results are shown in Figure 6 and Table 1. The major evaluation metrics taken include accuracy and parameters, and the proposed model achieves relatively great results for both in comparison, with an accuracy of 95.3% and parameters of only 1.21M, which reflect the effectiveness and efficiency of the proposed MSA-STGCN. Due to the limited modeling capability of the compared baseline model, it lacks consideration of the spatiotemporal dependencies between skeleton sequences, whereas the proposed model can obtain the long and short temporal dependencies between each frame's articulation points by combining multi-scale and channel attention mechanisms in spatio-temporal modeling. As a result, the proposed model shows a significant improvement in recognition accuracy compared with existing GCNs, and it improves by 2.1% compared with the best 2s-AGCN. Due to the multi-branching structure of the model in both temporal and spatial dimensions, and the eventual aggregation of multi-scale information, the number of parameters of the proposed model is substantially reduced. This effectively validates the accuracy and computational cost advantages of the model proposed for violent action recognition tasks.

TABLE 1 Comparison of different algorithms on Filtered NTU RGB+D dataset.

Methods	Accuracy (%)	Params (M)
Two-Stream CNN	93.4	1.53
HCN	92.7	1.03
ST-GCN	89.3	3.10
AS-GCN	89.1	9.50
J-AGCN	92.6	6.94
B-AGCN	91.1	6.94
2s-AGCN	93.2	6.94
Dynamic GCN	94.0	3.72
Ours	**95.3**	**1.21**

The bold values indicate the results of our proposed method (MSA-STGCN).

The main indicators of evaluation include accuracy and the number of parameters. The compared baseline models have limited modeling capability and lack the consideration of spatiotemporal dependencies among skeleton sequences, while the proposed model can obtain the long-term dependencies of an active state by combining multi-scale and channel attention mechanisms in the spatiotemporal modeling. Therefore, the proposed model has a significant improvement in recognition accuracy compared with the baseline model, which has improved by 2.1% compared with the best 2s-AGCN (Shi et al., 2019b). The proposed multi-information flow fusion method could fully exploit the specific relationships of the original data to further improves the recognition performance. The number of parameters of the proposed model can be reduced to 1.21M due to the multi-branch structure of the model in time and space dimensions, which effectively validates the accuracy and computational cost advantages.

Meanwhile, the 10 types of actions on the Filtered NTU RGB+D dataset: punching, kicking, pushing, slapping, pointing, hugging, giving, touching, handshaking, and walking were recognized, and the results are shown in Table 2. The recognition accuracy of these 10 types of actions were 91.1, 96.5, 94.7, 91.0, 93.6, 96.8, 91.5, 90.2, 96.0, and 98.6%, respectively. Normalized confusion matrix of 10 types of human action as shown in Figure 7, which illustrates that the method can be applied to violence recognition tasks in practical applications.

To further validate the generalization capability of the proposed recognition framework, we further conduct experiment on the Kinetics Skeleton 400 dataset, and Table 3 shown the results of the comparison experiments with ST-GCN, AS-GCN, ST-GR and 2s-AGCN. It can be seen that the proposed model achieves 36.2 and 58.5% accuracy in Top-1 and Top-5, respectively, which are still significant improvements compared to some of the baseline models. The results demonstrate that the proposed model can capture more features by combining multi-scale attention mechanisms, which can effectively identify more details in multi-frame skeleton sequences.

TABLE 2 Comparison of recognition results for 10 types of human action on the Filtered NTU RGB+D dataset.

Classes	Samples	True	Accuracy (%)
Punching	271	247	91.1
Kicking	260	251	96.5
Pushing	281	266	94.7
Slapping	278	253	91.0
Pointing	266	249	93.6
Hugging	278	269	96.8
Giving	281	257	91.5
Touching	287	259	90.2
Handshaking	273	262	96.0
Walking	277	273	98.6

4.3. Ablation study and discussion

4.3.1. Attention mechanism

This part mainly verifies the effectiveness of the attention mechanism proposed in the recognition framework by inserting the attention mechanism in the spatial dimensional to graph convolution network (ASGCN), and the experimental results are shown in Table 4. Firstly, the input skeleton sequences were tested for joints and bones in the spatial graph convolution layer (SGCN) without the SE Block, which was represented by J-ASGCN w/o SE and B-ASGCN w/o SE, respectively. Then, the results of the two data streams are fused and represented by ASGCN w/o SE. Finally, the SE Block attention

TABLE 3 Comparison of different algorithms on Kinetics Skeleton 400 dataset.

Methods	Top-1(%)	Top-5(%)
ST-GCN	30.7	52.8
AS-GCN	34.8	56.5
ST-GR	33.6	56.1
2s-AGCN	36.1	**58.7**
Ours	**36.2**	58.5

The bold values indicate the best accuracy.

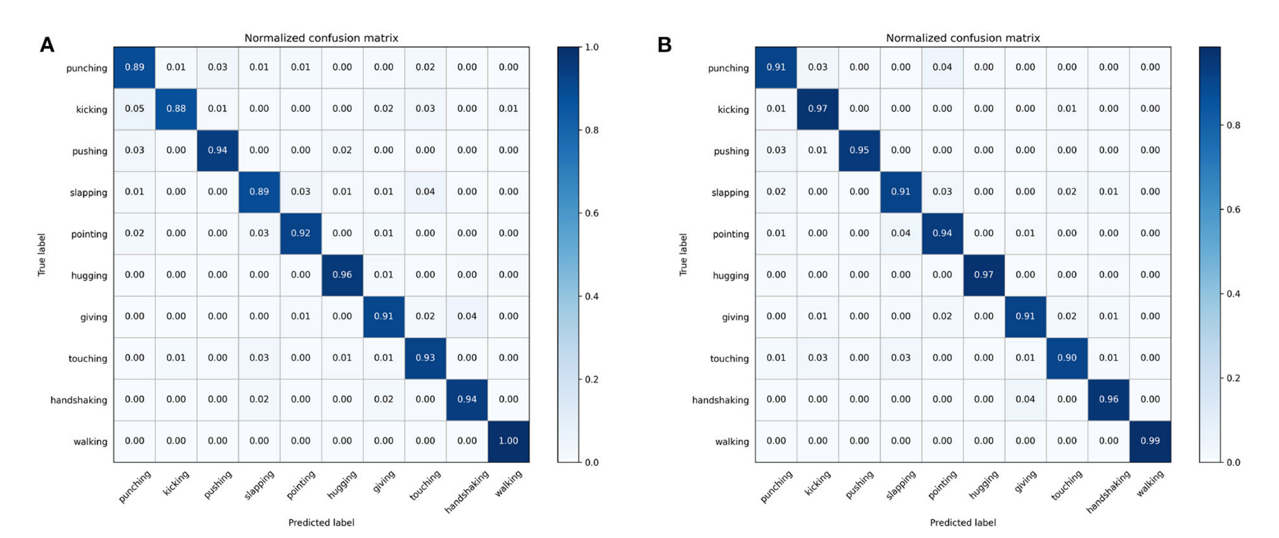


FIGURE 7
Normalized confusion matrix of 10 types of human action. **(A)** Confusion matrix of 2s-AGCN model. **(B)** Confusion matrix of our model.

mechanism is introduced in SGCN, and the model with the nodal position as input is represented by J-ASGCN, and the model with the skeletal vector as input is represented by B-ASGCN.

The variation accuracy of networks and the loss function values during the whole training process is shown in Figure 8. The recognition accuracy of J-ASGCN obtain 94.0% in the joint position information stream (increase of 0.4%), the B-ASGCN achieve 93.2% (increase of 0.1%) in the bone vector information stream, and the ASGCN achieved 94.9% (increase of 0.6%). Throughout the training process of the model, the accuracy of the test was improved rapidly in the early stage of the experiment, reaching about 85%, which is due to the high optimization efficiency of the proposed multi-scale spatial graph convolution. As the number of iterations increases, the final test accuracy and loss function converge very well, and the test accuracy and loss function curves are smoother in the later stage. Therefore, the attention mechanism SE Block does not play a significant role in this layer since the spatial feature extraction performance of the spatial map convolution layer itself is very robust. However, adding SE Block to our model can optimize the learning content and obtain more useful feature information, thus verifying the effectiveness of the method.

TABLE 4 Comparison of spatial graph convolution layer with and without SE block on the Filtered NTU RGB+D dataset.

Methods	Accuracy (%)
J-ASGCN w/o SE	93.6
B-ASGCN w/o SE	93.1
ASGCN w/o SE	94.3
J-ASGCN	94.0
B-ASGCN	93.2
ASGCN	**94.9**

The bold values indicate the accuracy of the model incorporating the attention mechanism.

4.3.2. Hybrid dilation convolution

Without pooling loss, the dilation convolution can increase the perceptual field of the feature map so that the output of each convolution contains a larger range of feature information. In this paper, we consider obtaining different sizes of perceptual fields in the temporal dimension to achieve a multi-scale fusion training network. To verify this idea, firstly, we compare the convolution rates of different sizes of voids, which are set to 1, 2, and 3, and the corresponding accuracy rates are 93.1, 93.2, and 93.5 respectively, as shown in Table 5. It is obvious

TABLE 5 Accuracy comparison of different dilated convolution rates used in temporal graph convolution layer on the Filtered NTU RGB+D dataset.

Methods	Accuracy (%)
MS-TCN(dilate rate = 1)	93.1
MS-TCN(dilate rate = 2)	93.2
MS-TCN(dilate rate = 3)	93.5
MS-TCN(HDC)	**94.0**

The bold values indicate the accuracy of the model using hybrid ablation convolution.

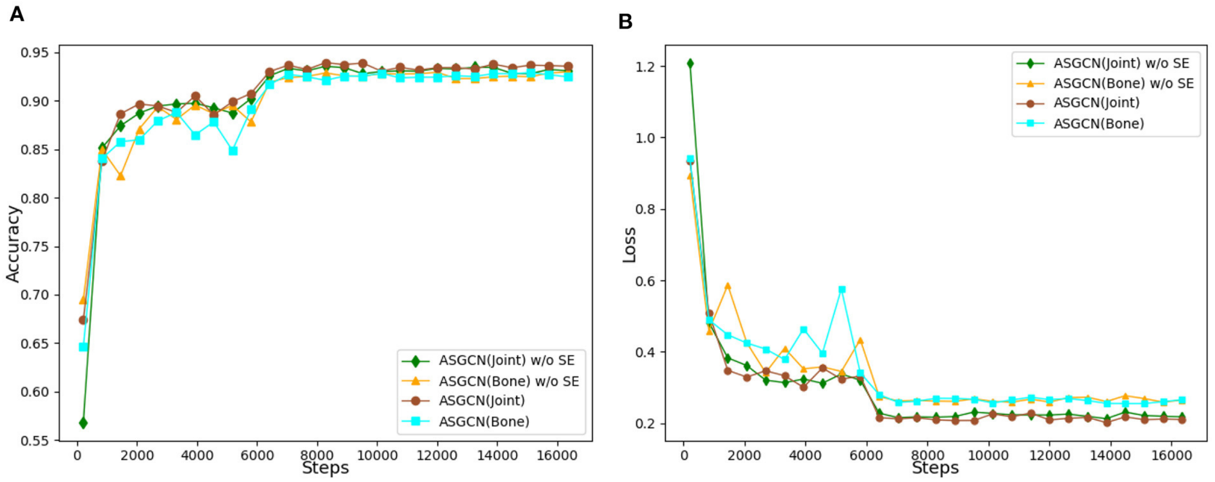


FIGURE 8
(A) Accuracy comparison of spatial graph convolution layer with or without SE block. (B) Loss function comparison of spatial graph convolution layer with or without SE block.

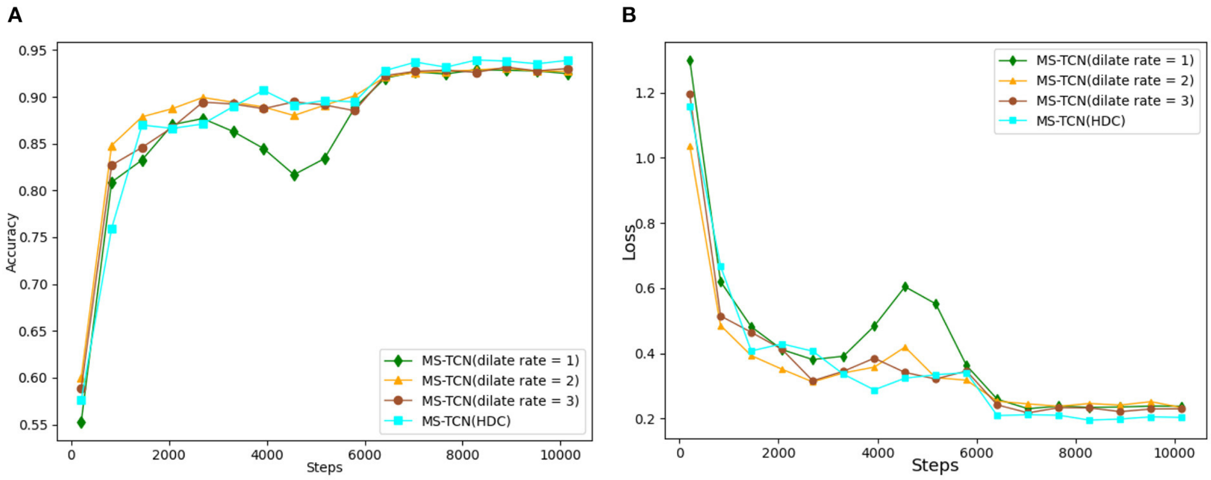


FIGURE 9
(A) Comparison of recognition accuracy with different dilated convolution rates. **(B)** Comparison of loss function with different dilated convolution rates.

that the accuracy of the model recognition is in a stable state with the increase of the hole convolution rate, which is not a very good training effect. Considering that the increase in the convolution rate of the dilation will bring about a grid effect, which will lead to the loss of continuity of a certain part of the feature information, and even, probably, the important feature information as well. Therefore, this paper solves the problem of discontinuity in the convolution kernel by designing a hybrid dilation convolution (HDC) form of temporal map convolution network, represented by MS-TCN(HDC). Finally, the accuracy of the MS-TCN(HDC) model reached 94.0% by fusing the hybrid dilation convolution form with different dilation rates.

The variation in the test accuracy of each network and the variation loss throughout the training process is shown in Figure 9. In the early stage of the experiment, the speed of convergence of the loss function increased slightly with the increase of the hole convolution rate, and the speed of test accuracy also increased. By adjusting the dilation convolution rate, the scale of the model is increased and the parameters of the network are changed, thus slightly improving the optimization efficiency of the network in the early stage of training. As the number of iterations increases, the final validation accuracy increases with the increase of the dilation convolution rate, and the training loss function achieves good convergence and a smoother curve in the later stages of training. The experimental results verify that the graph convolution network model constructed in the form of hybrid dilation convolution can learn more time-domain feature information at multiple scales compared with single dilation convolution.

TABLE 6 Accuracy comparison of different data stream recognition on the Filtered NTU RGB+D dataset.

Models	Accuracy (%)
J-MSAGCN	94.0
B-MSAGCN	93.2
J-M-MSAGCN	92.1
B-M-MSAGCN	93.4
MS-AGCN(fusion)	**95.3**

The bold values indicate the accuracy using multi-stream fusion.

4.3.3. Multi-stream fusion

Finally, the proposed multi-stream model incorporating joint position information, bone vector information, joint motion information, and bone motion information was tested and the experimental results are shown in Table 6. As for the input models of node position information, bone vector information, node motion information, and bone motion information, the corresponding accuracy rates were 94.0, 93.2, 92.1, and 93.4% for J-MSAGCN, B-MSAGCN, J-M-MSAGCN, and B-M-MSAGCN, respectively. The accuracy of MS-AGCN with a multi-stream fusion model could reach 95.3%.

During the whole training process, the variation in the accuracy of each network and the variation loss are shown in Figure 10. As the number of experimental iterations increased, the accuracy of the original joint position information stream increased slightly faster than the other three data streams in the early stage of the experiment, and the loss function also converged faster. This indicates that the original joint position

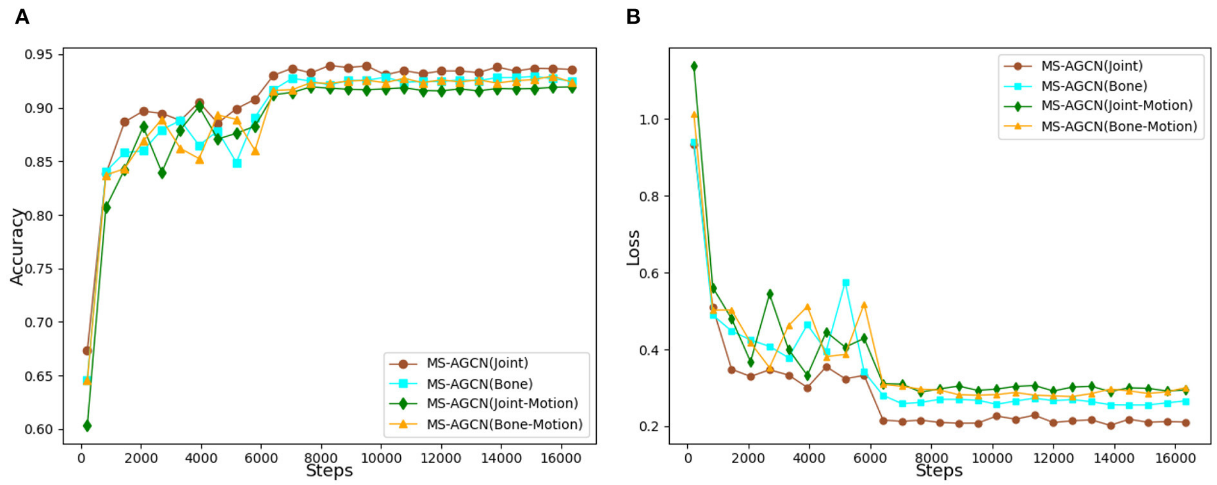


FIGURE 10
(A) Comparison of recognition accuracy of different data streams. **(B)** Comparison of loss functions of different data streams.

plays an important role in characterizing the movement state, while the accuracy of the other streams is increased by 1.3%, which suggests that by calculating the bone vector information, joint point motion information, and bone motion information, a higher weight is given to the more variable streams, thus enhancing the overall model's characterization of the movement. The experimental results show that the accuracy of the multi-stream fusion method is significantly higher than that of the single-stream method. In particular, the accuracy of the multi-stream fusion method has improved relative to the performance of the joint point information stream method. This shows that the skeleton sequence data can be extracted from different angles and the final fusion output can be used to fully characterize the action features.

5. Conclusion

In this paper, we design a novel spatiotemporal graph convolution network with attention mechanism to combine multi-stream skeleton features for human violence recognition. The proposed MSA-STGCN utilizes MSA-SGCN and MS-TGCN to learn spatial and temporal information from four types of skeleton data, respectively, and then a average features fusion mechnism is used to implement violence action classification. Compared with other traditional GCNs, the proposed MSA-STGCN achieves 95.3% accuracy on the Filtered NTU RGB+D dataset with only 1.21M model parameters, and the accuracy of Top-1 and Top-5 reached 36.2 and 58.5% on the Kinetics Skeleton 400 dataset, respectively. The experimental results demonstrate that the effectiveness of MSA-SGCN and MS-TGCN in the proposed MSA-STGCN recognition framework. Compared with the other state-of-the-arts, our framework consistently improves the recognition performance on two large skeleton datasets. In the future, more effective fusion and combining strategies that can help to obtain more distinctive complementary features from multimodal data such as RGB and depth sequences. Another future work is to expand more challenging datasets in order to enhance the generalization capability of the model and design RNN skeleton-based framework to learn the spatiotemporal features to improve recognition performance.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

ZR and QZ: conceptualization. HYa: methodology and validation. ZR and HYu: software and writing—review and editing. QZ: formal analysis and investigation. HYu: resources and funding acquisition. QZ and ZZ: data curation and visualization. HYa and ZR: writing—original draft preparation. ZR: supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the Dongguan Science and Technology Special Commissioner Project (No.

20221800500362), Dongguan Social Development Science and Technology Project (No. 20211800904722), the Key Project of Science and Technology Innovation 2030 supported by the Ministry of Science and Technology of China (No. 2018AAA0101301), the National Natural Science Foundation of China (No. 61972090), the Key Projects of Artificial Intelligence of High School in Guangdong Province (No. 2019KZDZX1011), Innovation Project of High School in Guangdong Province (No. 2018KTSCX314), and the Special Project in Key Fields of Guangdong Colleges and Universities (No. 2021ZDZX3007).

Acknowledgments

The authors thank everyone who contributed to this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Caetano, C., Brémond, F., and Schwartz, W. R. (2019a). "Skeleton image representation for 3D action recognition based on tree structure and reference joints," in *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (Rio de Janeiro), 16–23.

Caetano, C., Sena de Souza, J., dos Santos, J., and Schwartz, W. (2019b). "Skelemotion: a new representation of skeleton joint sequences based on motion information for 3D action recognition," in *2019 IEEE International Conference on Advanced Video and Signal-based Surveillance* (Taipei: IEEE), 1–8.

Cao, C., Zhang, Y., Zhang, C., and Lu, H. (2018). Body joint guided 3-d deep convolutional descriptors for action recognition. *IEEE Trans. Cybern.* 48, 1095–1108. doi: 10.1109/TCYB.2017.2756840

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2021). Openpose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 172–186. doi: 10.1109/TPAMI.2019.2929257

Carreira, J., and Zisserman, A. (2017). "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 4724–4733.

Chen, Y., Zhang, Z., Yuan, C., Li, B., Deng, Y., and Hu, W. (2021). "Channel-wise topology refinement graph convolution for skeleton-based action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (IEEE), 13359–13368.

Cheng, J., Ren, Z., Zhang, Q., Gao, X., and Hao, F. (2021). Cross-modality compensation convolutional neural networks for RGB-D action recognition. *IEEE Trans. Circ. Syst. Video Technol.* 32, 1498–1509. doi: 10.1109/TCSVT.2021.3076165

Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., and Lu, H. (2020). "Skeleton-based action recognition with shift graph convolutional network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 183–192.

Chi, H.-G., Ha, M. H., Chi, S., Lee, S. W., Huang, Q., and Ramani, K. (2022). "Infogcn: representation learning for human skeleton-based action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 20186–20196.

Degardin, B., Lopes, V., and Proença, H. (2021). Regina–reasoning graph convolutional networks in human action recognition. *IEEE Trans. Inf. Forensics Security* 16, 5442–5451. doi: 10.1109/TIFS.2021.3130437

Gadelkarim, M., Khodier, M., and Gomaa, W. (2022). "Violence detection and recognition from diverse video sources," in *2022 International Joint Conference on Neural Networks (IJCNN)* (Padua), 1–8.

Gao, X., Li, K., Zhang, Y., Miao, Q., Sheng, L., Xie, J., et al. (2019). "3D skeleton-based video action recognition by graph convolution network," in *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)* (Tianjin: IEEE), 500–501.

Hou, Y., Li, Z., Wang, P., and Li, W. (2018). Skeleton optical spectra-based action recognition using convolutional neural networks. *IEEE Trans. Circ. Syst. Video Technol.* 28, 807–811. doi: 10.1109/TCSVT.2016.2628339

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2020). Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2011–2023. doi: 10.1109/TPAMI.2019.2913372

Huang, Z., Shen, X., Tian, X., Li, H., Huang, J., and Hua, X.-S. (2020). "Spatio-temporal inception graph convolutional networks for skeleton-based action recognition," in *Proceedings of the 28th ACM International Conference on Multimedia, MM '20* (New York, NY: Association for Computing Machinery), 2122–2130.

Ji, Y., Yang, Y., Shen, F., Shen, H. T., and Zheng, W. S. (2021). Arbitrary-view human action recognition: a varying-view RGB-D action dataset. *IEEE Trans. Circ. Syst. Video Technol.* 31, 289–300. doi: 10.1109/TCSVT.2020.2975845

Jiang, X., Xu, K., and Sun, T. (2020). Action recognition scheme based on skeleton representation with DS-LSTM network. *IEEE Trans. Circ. Syst. Video Technol.* 30, 2129–2140. doi: 10.1109/TCSVT.2019.2914137

Kim, T. S., and Reiter, A. (2017). "Interpretable 3D human action analysis with temporal convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Honolulu, HI: IEEE), 1623–1631.

Li, B., Li, X., Zhang, Z., and Wu, F. (2019). Spatio-temporal graph routing for skeleton-based action recognition. *Proc. AAAI Conf. Artif. Intell.* 33, 8561–8568. doi: 10.1609/aaai.v33i01.33018561

Li, C., Hou, Y., Wang, P., and Li, W. (2019). Multiview-based 3-D action recognition using deep networks. *IEEE Trans. Hum. Mach. Syst.* 49, 95–104. doi: 10.1109/THMS.2018.2883001

Li, C., Xie, C., Zhang, B., Han, J., Zhen, X., and Chen, J. (2022). Memory attention networks for skeleton-based action recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 33, 4800–4814. doi: 10.1109/TNNLS.2021.3061115

Li, C., Zhang, Q., Di, X., and Shiliang, P. (2018). "Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation," in *IJCAI'18: Proceedings of the 27th International Joint Conference on Artificial Intelligence* (Stockholm), 782–796.

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). "Actional-structural graph convolutional networks for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 3590–3598.

Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2022). Symbiotic graph neural networks for 3D skeleton-based human action recognition and motion prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 3316–3333. doi: 10.1109/TPAMI.2021.3053765

Li, S., Li, W., Cook, C., Zhu, C., and Gao, Y. (2018). "Independently recurrent neural network (indrnn): building a longer and deeper RNN," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 5457–5466.

Li, T., Ke, Q., Rahmani, H., Ho, R. E., Ding, H., and Liu, J. (2021). "Else-net: elastic semantic network for continual action recognition from skeleton data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 13434–13443.

Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.-Y., and Kot, A. C. (2020). NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 2684–2701. doi: 10.1109/TPAMI.2019.2916873

Liu, J., Shahroudy, A., Xu, D., Kot, A. C., and Wang, G. (2018). Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 3007–3021. doi: 10.1109/TPAMI.2017.2771306

Liu, Z., Zhang, H., Chen, Z., Wang, Z., and Ouyang, W. (2020). "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 140–149.

Ople, J. J. M., Yeh, P.-Y., Sun, S.-W., Tsai, I.-T., and Hua, K.-L. (2020). Multi-scale neural network with dilated convolutions for image deblurring. *IEEE Access* 8, 53942–53952. doi: 10.1109/ACCESS.2020.2980996

Ramzan, M., Abid, A., Khan, H. U., Awan, S. M., Ismail, A., Ahmed, M., et al. (2019). A review on state-of-the-art violence detection techniques. *IEEE Access* 7, 107560–107575. doi: 10.1109/ACCESS.2019.2932114

Senst, T., Eiselein, V., Kuhn, A., and Sikora, T. (2017). Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation. *IEEE Trans. Inf. Forensics Security* 12, 2945–2956. doi: 10.1109/TIFS.2017.2725820

Serrano, I., Deniz, O., Espinosa-Aranda, J. L., and Bueno, G. (2018). Fight recognition in video using hough forests and 2D convolutional neural network. *IEEE Trans. Image Process.* 27, 4787–4797. doi: 10.1109/TIP.2018.2845742

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019a). "Skeleton-based action recognition with directed graph neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 7904–7913.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2019b). "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 12018–12027.

Shi, L., Zhang, Y., Cheng, J., and Lu, H. (2020). Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. *IEEE Trans. Image Process.* 29, 9532–9545. doi: 10.1109/TIP.2020.3028207

Shu, X., Zhang, L., Sun, Y., and Tang, J. (2021). Host-parasite: Graph LSTM-in-LSTM for group activity recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 32, 663–674. doi: 10.1109/TNNLS.2020.2978942

Si, C., Chen, W., Wang, W., Wang, L., and Tan, T. (2019). "An attention enhanced graph convolutional lstm network for skeleton-based action recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 1227–1236.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2017). "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," in *AAAI Conference on Artificial Intelligence* (San Francisco, CA: AAAI), 1–7.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J. (2018). Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. *IEEE Trans. Image Process.* 27, 3459–3471. doi: 10.1109/TIP.2018.2818328

Song, Y.-F., Zhang, Z., Shan, C., and Wang, L. (2022). Constructing stronger and faster baselines for skeleton-based action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* p. 1–15. doi: 10.1109/TPAMI.2022.3157033

Tu, Z., Zhang, J., Li, H., Chen, Y., and Yuan, J. (2022). Joint-bone fusion graph convolutional network for semi-supervised skeleton action recognition. *IEEE Trans. Multimedia.* p. 1–13. doi: 10.1109/TMM.2022.3168137

Wang, H., and Wang, L. (2017). "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI: IEEE), 3633–3642.

Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., et al. (2018). "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Lake Tahoe, NV: IEEE), 1451–1460.

Wu, C., Wu, X.-J., and Kittler, J. (2019). "Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (Seou: IEEE), 1740–1748.

Xu, Y., Cheng, J., Wang, L., Xia, H., Liu, F., and Tao, D. (2018). Ensemble one-dimensional convolution neural networks for skeleton-based action recognition. *IEEE Signal Process. Lett.* 25, 1044–1048. doi: 10.1109/LSP.2018.2841649

Yan, S., Xiong, Y., and Lin, D. (2018). "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *2018 AAAI Conference on Artificial Intelligence* (New Orleans, LO: AAAI), 1–10.

Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., and Zheng, N. (2020). "Semantics-guided neural networks for efficient skeleton-based human action recognition," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 1109–1118.

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 15 December 2022
DOI 10.3389/fnbot.2022.993936

OPEN ACCESS

EDITED BY
Huiyu Zhou,
University of Leicester, United Kingdom

REVIEWED BY
Tharun Kumar Reddy,
Indian Institute of Technology Roorkee, India
Jingjing Luo,
Fudan University, China

*CORRESPONDENCE
Jinqiang Cui
cuijq@pcl.ac.cn

RECEIVED 14 July 2022
ACCEPTED 22 November 2022
PUBLISHED 15 December 2022

CITATION
Liu Y, Cui J and Liang W (2022) A hybrid learning-based stochastic noise eliminating method with attention-Conv-LSTM network for low-cost MEMS gyroscope. *Front. Neurorobot.* 16:993936. doi: 10.3389/fnbot.2022.993936

A hybrid learning-based stochastic noise eliminating method with attention-Conv-LSTM network for low-cost MEMS gyroscope

Yaohua Liu[1,2,3], Jinqiang Cui[3]* and Wei Liang[1,2]

[1]School of Nano-Tech and Nano-Bionics, University of Science and Technology of China, Hefei, China, [2]Institute of Nano-Tech and Nano-Bionics, Chinese Academy of Sciences, Suzhou, China, [3]Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, China

Low-cost inertial measurement units (IMUs) based on microelectromechanical system (MEMS) have been widely used in self-localization for autonomous robots due to their small size and low power consumption. However, the low-cost MEMS IMUs often suffer from complex, non-linear, time-varying noise and errors. In order to improve the low-cost MEMS IMU gyroscope performance, a data-driven denoising method is proposed in this paper to reduce stochastic errors. Specifically, an attention-based learning architecture of convolutional neural network (CNN) and long short-term memory (LSTM) is employed to extract the local features and learn the temporal correlation from the MEMS IMU gyroscope raw signals. The attention mechanism is appropriately designed to distinguish the importance of the features at different times by automatically assigning different weights. Numerical real field, datasets and ablation experiments are performed to evaluate the effectiveness of the proposed algorithm. Compared to the raw gyroscope data, the experimental results demonstrate that the average errors of bias instability and angle random walk are reduced by 57.1 and 66.7%.

KEYWORDS
MEMS IMU, deep learning, noise reduction, inertial navigation, random noise

1. Introduction

Recently, with the development of the microelectromechanical system (MEMS) and artificial intelligence (AI), the low-cost MEMS inertial measurement units (IMUs) are essential for many applications, such as unmanned aerial vehicles, autonomous driving, mobile robots, etc. IMUs consist of gyroscopes that measure angular velocities and accelerometers that measure the accelerations of moving vehicles. The IMUs can provide the entire attitude, velocity, and position information through the integral operation. However, the measurement errors will accumulate over time due to the bias error instability and stochastic noise in raw IMU data.

Specifically, the position error of inertial navigation diverges with the second power of accelerometer bias drift and time, and diverges with the third power of gyroscope bias drift and time. Therefore, modeling or denoising the low-cost MEMS IMUs is crucial to improving the inertial navigation system (INS) performance.

In inertial navigation, the errors contained in the MEMS IMU raw signals can be divided into two parts: deterministic and stochastic errors. The deterministic error part mainly includes the scale factor error and axes misalignment error, which can be calibrated or quantified by equipment such as a 3-axis turntable. While the stochastic errors consist of bias error and noise, they are hard to calibrate due to their time-varying characteristic. Thus, the stochastic errors are also a vital issue of the INS errors divergence. In order to identify and model the stochastic error, researchers have proposed many representative denoising techniques for MEMS IMUs, which can be inclusively divided into conventional signal processing methods and recent learning-based methods. Auto Regressive Moving Average Method (ARMA) (Song et al., 2018), Allan Variance (Zhang et al., 2018), Kalman Filter (Zhang et al., 2016), and Wavelet Transformation (WT) (Yuan et al., 2015) are the representative signal processing methods. The ARMA method is mainly used to analyze and study a group of stochastic data arranged in sequence. It establishes mathematical models of various orders according to different error sequences. However, this method cannot identify stochastic errors one by one, and it is difficult to distinguish the error sources of stochastic errors. Allan Variance can identify various stochastic errors and separate them into five parts: quantization noise, angular random walk, bias instability, rate random walk, and rate ramp. Thus, the advantage of the Allan Variance is that it can draw a double logarithmic curve to connect the time and frequency domains and visually observe the stochastic errors quantitatively. When the amount of data is large enough, the drawn double logarithmic curve is more intuitive and straightforward (El-Sheimy et al., 2007). Kalman Filter is an efficient linear quadratic estimator which can estimate gyroscope output angular velocity *via* a series of observed measurements with noise (Cai et al., 2018). Since the natural MEMS IMU error system is usually too complex to build an accurate mathematics model, the Kalman filter has a poor performance in estimation accuracy. Among the signal processing methods, the Wavelet Transform method is currently most popular for reducing the high-frequency part of the gyroscope error. However, it is hard to remove the low-frequency errors (Ding et al., 2021). The the signal processing algorithms can only reduce part of the MEMS gyroscope stochastic error, and the unsatisfactory suppression of the stochastic errors will cause the failure of inertial navigation in a short time.

Other learning-based approaches are proposed to improve the traditional statistical algorithms, such as support vector machine (SVM) and neural networks, all of which obtain better denoising results than conventional signal processing methods (Leung et al., 2001; Shiau et al., 2011; Bhatt et al., 2012). In Zhang and Yang (2012), the SVM is utilized to model and compensate for the angular rate error of MEMS gyroscope MG31-300, which indicates that the SVM model has high precision and good generalization ability. A basis function neural network is adopted to predict the noisy chaotic time series due to its non-linear, adaptive, and self-learning characteristics (Leung et al., 2001). Gonzalez and Catania (2019) proposed a rigorous analysis of the viability of the Time Delayed Multiple Linear regression techniques for reducing white noise in the MEMS IMU. Their advantages rely on their ability to identify complex patterns by learning high-level data features. However, almost all of the above methods are based on a static model, which models only the current and past one-step angular velocity information and can not store more past gyroscope dynamic information. It is known that gyroscope data is time serial data in which the history error will affect the current measurement value.

In recent years, deep learning has achieved outstanding performances in computer vision (Han et al., 2020) and natural language processing (Koroteev, 2021) due to their powerful non-linear modeling and feature representation. Some researchers have introduced deep learning into the inertial odometer, such as OriNet (Esfahani et al., 2019), IONet (Chen et al., 2018), TLIO (Liu et al., 2020), all of which obtained excellent localization performance than traditional methods. However, the use of deep learning technology to reduce MEMS IMU stochastic noise has just begun, and the published research results are still rare. In Jiang et al. (2018a), an recurrent neural network (RNN) variant simple recurrent unit (SRU-RNN) is employed in MEMS gyroscope raw signal denoising. The Allan variance tool is also used to compute the major error factors, i.e., quantization noise, angle random walk, and bias instability. However, RNN performs poorly in long sequences due to gradient disappearance and gradient explosion. To solve such problem, long short-term memory (LSTM) has been proposed (Graves et al., 2005; Sherstinsky, 2020), which can be used to denoise the MEMS gyroscope based on the current and previous angular velocities. In Jiang et al. (2018b), the LSTM is employed to filter the MEMS gyroscope outputs by treating the signals as time series. The results indicated that the denoising scheme effectively improves MEMS gyroscope accuracy. To further explore the effect of LSTM in denoising the MEMS gyroscope, some hybrid deep recurrent neural networks, including LSTM and gated recurrent unit (GRU), are evaluated for MEMS IMU with static and dynamic conditions (Han et al., 2021). The LSTM is also combined with the Kalman filter to estimate and compensate for the random drift of the MEMS gyroscope in real-time (Li et al., 2021; Zhu et al., 2021). It is noted that the RNN can learn the temporal correlation from the useful signals of the original data, but it cannot learn from the noisy components (Shiau et al., 2011). Thus, RNNs have a poor ability to extract the local features of MEMS gyroscope. To solve the problems, a convolutional neural network (CNN) is applied to reduce the attitude angle errors and achieve better denoising performances (Brossard et al., 2020). However, we focus on eliminating the stochastic noise in raw MEMS gyroscope data, rather than calibrating IMU error by reducing the attitude angle error.

As already discussed above, it can be seen that most eliminating MEMS gyroscope stochastic noise works (90%) are based on the RNNs; significantly, only one hybrid model with LSTM and GRU. None of the above methods can simultaneously extract the local features of the MEMS gyroscope and learn the long-range dependence. In addition, they can not explore different levels of the importance of gyroscope sequences at different times.

Therefore, this paper aims to develop a hybrid MEMS gyroscope denoising scheme based on Attention-CNN-LSTM (ACL) to eliminate the stochastic noise for angular velocity. Although there are similar hybrid models in other fields, such as stock prediction, we focus on MEMS gyroscope stochastic noise reduction, and there is no research on the hybrid denoising model so far. Specially, a one-dimensional CNN is adopted in the proposed ACL to extract local MEMS gyroscope features. The features are fed to the LSTM layer to mine the temporal features further and learn the long-term historical dependence. In order to improve computing efficiency, an attention mechanism is applied to distinguish the importance of MEMS gyroscope sequences at different times. The contributions of the paper are summarized as follows:

1. We develop a hybrid denoising model based on Conv-LSTM networks to capture the spatial-temporal feature of the MEMS gyroscope sequence. Unlike the existing RNN-based method for denoising gyroscopes, Conv-LSTM can capture the sectional features and learn long-range dependencies simultaneously, which is more efficient for mining the inherent characteristic of the gyroscope sequence.
2. We embed an attention mechanism for the Conv-LSTM model to automatically allocate different attention weights to a gyroscope sequence at different times, which can further improve the efficiency of the Conv-LSTM model.
3. A series of experiments are performed to verify the effectiveness of the proposed method. The experimental results demonstrate that the proposed model performs better than other gyroscope denoising methods.

The remainder of the paper is organized as follows. Section 2 explains the mathematical model of low-cost MEMS IMU in detail. Section 3 describes the process of establishing a denoising model based on ACL. Real field, datasets and ablation experiments and results analysis are discussed in Section 4. The conclusion is provided in Section 5.

2. The mathematical models of low-cost MEMS IMU

Low-cost MEMS IMUs are prone to various errors, which get more complex as the sensor price decreases. The errors limit the accuracy to which the observables can be measured. In this section, the output models of the MEMS IMUs are presented to analyze their error characteristics.

2.1. The errors of the low-cost MEMS IMUs

MEMS IMU contains two orthogonal sensor triads, one with three accelerometers and the other with three gyroscopes. Accelerometers measure linear motion in three orthogonal directions, whereas gyroscopes measure angular motion in three orthogonal directions. However, owing to the limitation of current MEMS manufacturing technology, the output of the MEMS IMU is affected by many error sources.

The general terms of repeatability, stability, and drift are usually considered to assess a MEMS IMU sensor for a particular application. The repeatability term represents the ability of a MEMS IMU to provide the same output for repeated applications of the same input. It refers to the maximum variation between repeated measurements in the same conditions over multiple runs. The stability term illustrates the ability of a MEMS IMU to provide the same output when measuring a constant input over a while. The term drift is often used to describe the change in the MEMS IMU measurement when there is no change in the input. Especially the MEMS IMU errors can be classified into two broad categories of deterministic and stochastic errors. Deterministic errors mainly include systematic bias offset, scale factor error, non-linearity, non-orthogonality error, and misalignment error. Most of the deterministic errors can only be found in dynamic environments, and can be compensated by laboratory calibration process. Low-cost MEMS IMUs suffer from various stochastic errors, which are usually modeled stochastically to mitigate their effects. In general, the stochastic errors of Low-cost MEMS IMU can be divided into run-to-run bias offset, bias drift, scale factor instability, and white noise. Any above stochastic errors will cause the navigation results (attitude, velocity, and position) to diverge rapidly in the inertial navigation system. Therefore, it is fundamental to suppress the stochastic errors of the low-cost MEMS IMUs.

The initial error of IMU is relatively tiny, but as time goes on, the position and speed position calculated by the inertial navigation algorithm will become larger and larger. The position of inertial navigation can be expressed as follows,

$$\delta r_N = \delta r_{N,0} + \delta v_{N,0} \cdot t + \frac{1}{2}(g \cdot \delta\theta_0 + b_{aN})t^2 + \frac{1}{6}(g \cdot b_{gE})t^3 \quad (1)$$

where δr_N is north position error, $\delta r_{N,0}$, $\delta v_{N,0}$, and $\delta\theta_0$ represent north position error, velocity error and yaw angle error at initial time, respectively. b_{aN} and b_{gE} are the bias error of the accelerometer in the north direction and gyroscope in the east direction. g is local gravity, and t is inertial navigation time. It can be seen that the bias drift of the accelerometer will cause position error to diverge with the second power of time, and the bias drift of the gyroscope will cause position error to diverge with the third power of time. If the MEMS IMU is not denoised well, the position information calculated by the MEMS IMU will not be used for navigation.

2.2. The output model base on low-cost MEMS IMUs

In the field of inertial navigation, the output model based on low-cost MEMS IMUs includes the angular rate model and the specific force model, i.e., the measurements of the gyroscope and accelerometer, respectively. Measurements of angular rate can be expressed as follows:

$$\tilde{\omega}_{ib}^{b} = \omega_{ib}^{b} + b_g + S_g\omega_{ib}^{b} + N_g\omega_{ib}^{b} + \varepsilon_g \tag{2}$$

where ω_{ib}^{b} is the real values of the angular velocity in the body frame b relative to the inertial frame i, and $\tilde{\omega}_{ib}^{b}$ is the output values of the gyroscope. Furthermore, b_g, ε_g, S_g, and N_g are the gyroscope instrument bias vector, noise vector, scale factor matrix and non-orthogonality matrix, respectively. The bias vector is defined as the gyroscope's output when there is zero input. The noise vector is white noise, which can be caused by power sources but can also be intrinsic to semiconductor devices. The scale factor matrix reflects the deviation of the input-output gradient from unity. As the name suggests, non-orthogonality errors occur when any of the axes of the gyroscope triad depart from mutual orthogonality. The matrices N_g and S_g are given as,

$$N_g = \begin{bmatrix} 1 & \theta_{g,xy} & \theta_{g,xz} \\ \theta_{g,yx} & 1 & \theta_{g,yz} \\ \theta_{g,zx} & \theta_{g,zy} & 1 \end{bmatrix}, S_g = \begin{bmatrix} s_{g,x} & 0 & 0 \\ 0 & s_{g,y} & 0 \\ 0 & 0 & s_{g,z} \end{bmatrix} \tag{3}$$

where $\theta_{g,\cdot}$ are the small angles defining the misalignments between the different gyroscope axes and $s_{g,\cdot}$ are the scale factors for the three gyroscopes.

The attitude angular increment is obtained by integrating the measured value of the gyroscope, namely,

$$\begin{aligned} R(t) &= R(t-1)\exp(\theta_t) \\ \theta_t &= \tilde{\omega}_{ib}^{b}(t)dt \\ \exp(\theta_t) &= I + \tfrac{\sin\theta_t}{\theta_t}[\theta_t\times] + \tfrac{1-\cos\theta_t}{\theta_t^2}[\theta_t\times]^2 \end{aligned} \tag{4}$$

where $\tilde{\omega}_{ib}^{b}(t)$ is the output of the gyroscope and is also the angular velocity of the body frame b relative to the inertial frame i, $R(t)$ is the rotation matrix of the body frame b relative to the inertial frame i, $[\theta_t\times]$ is the antisymmetric matrix of θ_t, θ_t is attitude angles.

From Equation (2), S_g and N_g can be reduced by the calibration processing with a turntable. b_g and ε_g are hard to estimated by traditional method due to their time-varying characteristic. If the errors cannot be reduced, the errors will be transferred to the rotation matrix and they will accumulate over time according to Equation (2). Thus, our goal is to establish a denoising model based on deep learning to reduce b_g and ε_g. In other words, we use the deep learning model to denoise the gyroscope, reducing the errors of b_g and ε_g, so that the gyroscope measurements $\tilde{\omega}_{ib}^{b}$ are closer to the true value ω_{ib}^{b}, and the attitude angles θ_t can be estimated more accurately through Equation (2).

The output error model of the accelerometer is similar to those which characterize the gyroscope accuracy bias uncertainty, scale factor stability, and random noise. Measurement of the specific force can be modeled by the observation equation,

$$\tilde{f}^b = f^b + b_a + S_1 f + S_2 f^2 + N_a f + \delta g + \varepsilon_a \tag{5}$$

where $\tilde{f}^b$, f^b, b_a, δg, and ε_a are the vectors of the accelerometer measurement, the true specific force, the accelerometer instrument bias, the anomalous gravity and noise, respectively. Similar to the gyroscope, S_1, S_2, and N_a are the error matrices of linear scale factor, non-linear scale factor and non-orthogonality. The matrices N_a, S_1, and S_2 are defined as follows,

$$\begin{aligned} N_a &= \begin{bmatrix} 1 & \theta_{a,xy} & \theta_{a,xz} \\ \theta_{a,yx} & 1 & \theta_{a,yz} \\ \theta_{a,zx} & \theta_{a,zy} & 1 \end{bmatrix}, S_1 = \begin{bmatrix} s_{1,x} & 0 & 0 \\ 0 & s_{1,y} & 0 \\ 0 & 0 & s_{1,z} \end{bmatrix}, \\ S_2 &= \begin{bmatrix} s_{2,x} & 0 & 0 \\ 0 & s_{2,y} & 0 \\ 0 & 0 & s_{2,z} \end{bmatrix} \end{aligned} \tag{6}$$

where $\theta_{a,*}$ are the small angles defining the misalignments between the different accelerometer axes and s are the scale factors for the three accelerometers.

3. MEMS IMU stochastic errors reduction method based on deep learning

In order to improve the accuracy of the low-cost MEMS IMUs, a hybrid deep learning model with attention-based CNN-LSTM networks is proposed to reduce stochastic errors. In this section, the network architecture is illustrated and the principles of CNN, LSTM and attention mechanism are also introduced.

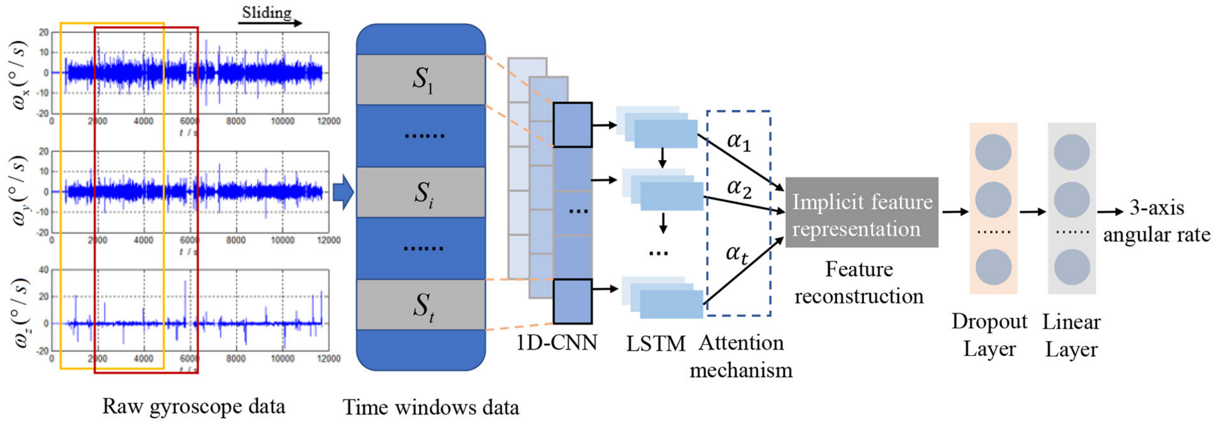


FIGURE 1
The architecture of the prediction model.

3.1. Network architecture

As illustrated in Figure 1, the deep learning model of denoising low-cost MEMS IMU, namely ACL, mainly consists of 1D-CNN layers, LSTM layers, and an attention mechanism. Since the error characteristics of MEMS gyroscope and accelerometer are similar, and gyroscope is essential for inertial navigation, we will take gyroscope data as an example to analyze the noise reduction process based on the proposed ACL model. The raw angular rate signals from the MEMS gyroscope sensors can be observed within the different time windows. Moreover, the i_{th} observed data S_i is fed into the 1D-CNN layers, which act as feature extractors to automatically obtain the local features and provide abstract representations of the input sensor data in the feature maps. So the noise reduction problem can be formulated as follows,

$$Denoised\_Gyro = ACL - NN(S_1, S_2, ..., S_k) \tag{7}$$

LSTM layers can further learn the long-term historical dependence from the results of the previous convolution output. Meanwhile, an attention mechanism is designed to explore different levels of the importance of gyroscope sequences at different times. A dropout layer is also applied to avoid overfitting. A linear layer is added to transform high dimension data as the output data dimension shape to predict low noise angular rate. Each module will be described in detail in the following subsections.

3.2. One dimensional convolutional neural network (1D-CNN)

1D-CNN is widely used in time series analysis, audio signal data with fixed length periods, natural language processing, etc. The angular velocities and accelerations of the vehicles measured by MEMS IMUs can be regarded as a kind of time-series sequence. For example, the gyroscope sequences of the i_{th} time window can be expressed,

$$S_i = [x_1, x_2, ..., x_l] \tag{8}$$

where l is window size and x_t represents the raw angular velocities from the gyroscope at time t.

The 1D-CNN is used to extract local error features from raw MEMS IMU data in our proposed method. The specific convolution operation is,

$$c_k = \mathrm{ReLU}(\omega_k * x + \mathrm{b}) \tag{9}$$

where c_k is the output feature map of the k_th kernel, ω_k and b are weight and deviation parameters. As shown in Figure 2, the convolution operator slides along the time direction and outputs the feature map. Since the 3-axis gyroscope data is fed to 1D-CNN in our model, the number of input channel is set 3. The gyroscope records the angular velocity of the carrier at each time, so the length of the gyroscope measurement sequence cannot be changed. To ensure that the input sequence and output sequence of the gyroscopes are the same length after 1D-CNN operation and reduce the model parameters, the convolution kernel size and the layer of 1D-CNN are set to 1. In order to improve the representation ability of extracted features, the output channel size of 1D-CNN is 256.

3.3. Long short term memory (LSTM)

RNN is a popular branch of the deep learning method, where the connections among nodes can form a directed graph along a sequence. Unlike feedforward neural networks, RNNs can use

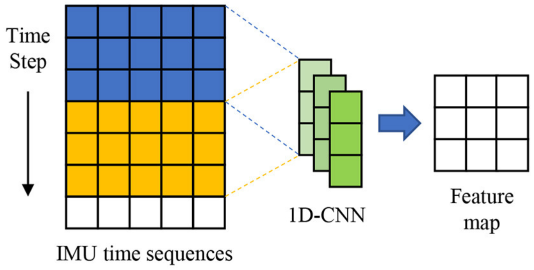


FIGURE 2
Illustration of the 1D convolution operation.

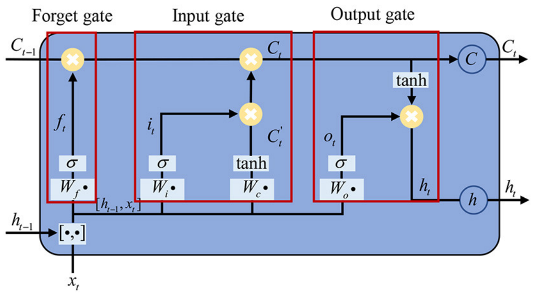


FIGURE 3
Illustration of the LSTM network.

their internal state (memory) to process sequences of inputs. However, it has a vanishing gradient problem that is unable to find an appropriate gradient in long-term memory (Gers et al., 2000; Sutskever et al., 2014).

An RNN composed of LSTM units is often called an LSTM network, which contains a cell, an input gate, an output gate, and a forgetting gate to avoid the vanishing gradient problem. An LSTM memory unit is shown in Figure 3. LSTM uses two gates to control the contents of the unit state C. One is the forgetting gate, which determines how much of the cell state in the previous moment C_{t-1} is retained in the current cell state C_t. The other one is the input gate, which determines the level of input of the current network X_t is saved to the cell state C_t. The LSTM NN uses the output gate to control the level of the unit state C_t sent to the current output h_t (Sak et al., 2014). The current input cell status $\tilde{C}_t$ can be calculated based on the previous output. The final LSTM output is determined by both the output gate and the unit state.

In our model, the LSTM input channel size is the same as the previous 1D-CNN output channel size, i.e., 256, and the output channel size is 128.

TABLE 1 The specifications of the AHRS380SA-200.

Gyroscope	Range ($^\circ/s$)	± 180
	Bias instability ($^\circ/hr$)	<10
	Angular random walk ($^\circ/\sqrt{hr}$)	<0.75
Accelerometer	Range (g)	± 4
	Bias instability (mg)	<0.02
	Velocity random walk ($m/s/\sqrt{hr}$)	<0.05
Physical	Size (mm)	41*48*22
	Weight (gm)	<30
Electrical	Output data rate (Hz)	2 to 100
	Input voltage (VDC)	9–32
	Power consumption (mW)	<350

3.4. Attention mechanism

The attention mechanism is a technique that mimics human cognitive attention by selectively ignoring part of the unimportant information and focusing on specific objects. It lays the groundwork for variants of subsequent attention mechanisms and has been successfully used in computer vision, recommendation systems, and translation (Bahdanau et al., 2014). In the context of neural networks, the attention mechanism can be regarded as a weight matrix. In other words, each input data have a corresponding weight value by assigning the attention degree, and the stronger the attention, the greater the weight.

As is known, the time sequence data of the MEMS gyroscope contain more complex temporal information. The error features information of the MEMS gyroscope computed by the LSTM at different times may influence the angular velocities differently. For example, the initial error at a time window will accumulate over time and have a greater impact than the error at the end of the time window. However, the standard LSTM cannot deal with the different important parts of the gyroscope sequence well. Therefore, soft attention (Zhao et al., 2020) is adopted to automatically distinguish different levels of importance of the error features at different times. The attention mechanism can be expressed as,

$$\alpha_i = \frac{\exp(s_i)}{\sum_{i=1}^{t-1} \exp(s_i)} \tag{10}$$

where α_i represents the importance of the $i_t h$ time window for MEMS gyroscope sequence prediction, and the score s_i is the attention weight.

4. Experimental results and analysis

The real field, dataset and ablation tests are performed in this section to evaluate the proposed algorithm. Allan

TABLE 2 Network structure and training hyperparameters tuning.

	Learning rate	Epoch number	Batch size	Dropout	CNN layer	CNN output channel size	LSTM layer	LSTM output channel size
		100						
Range	1e-4	150	64	0.1	1,2	64,128	1,2	64,128
	1e-3	200	128	0.2	3,4	256,512	3,4	256,512
	1e-2	250	256	0.3				
Value	1e-4	150	64	0.2	1	256	1	128

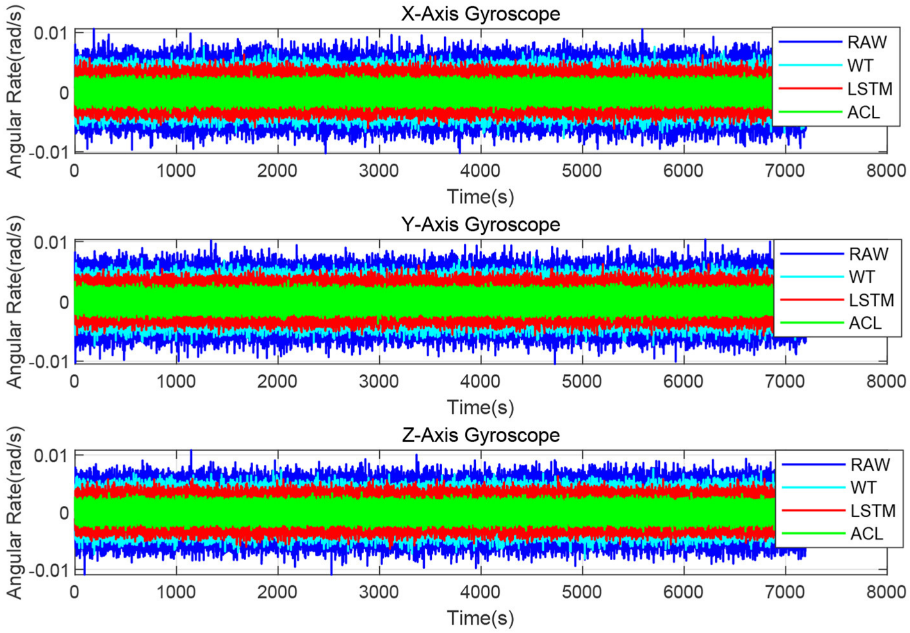


FIGURE 4
The denoised and raw signals comparison for the 3-axis gyroscope of AHRS380SA.

variance is used to quantitatively analyze the stochastic noise reduction effects.

4.1. Real field tests

In order to verify the performance of our method, a popular low-cost MEMS IMU AHRS380SA-200 manufactured by ACEINNA company is employed in this study. The IMU is composed of 3-orthogonal gyroscopes and 3-orthogonal accelerometers. As listed in Table 1, the full measurement range, maximum bias instability and angle random walk of the AHRS380SA-200 are $\pm 180^\circ/s$, $10^\circ/h$ and $0.75^\circ/\sqrt{hr}$.

During the raw signal collecting, the AHRS380SA-200 is placed on the table statically, and the sampling frequency is set to 100 Hz at room temperature. A computer-installed data acquisition software retrieved the raw signals *via* a MOXA USB to RS-232 data conversion cable. The Pytorch 1.8 is used as the deep learning framework tool, and the computer used in the experiment is configured as Intel Corei7-6700 3.4 GHz, 16GB RAM, RTX2080ti GPU. Two hours of gyroscope output data is used to train the model. In contrast, the same raw data length is adopted to evaluate the model's performance and tune the model parameters. The Allan Variance method is selected to analyze and describe the composition of the gyroscope noise contained in the raw output signals, which is a time-domain analysis technique originally designed for characterizing noise and stability in clock systems (Woodman, 2007). For LSTM networks, the sequence length can determine how much context information is sent to the model each time. Considering the IMU sampling rate, a window size of 100 is applied to the IMU sequence data to reduce memory. In order to reduce

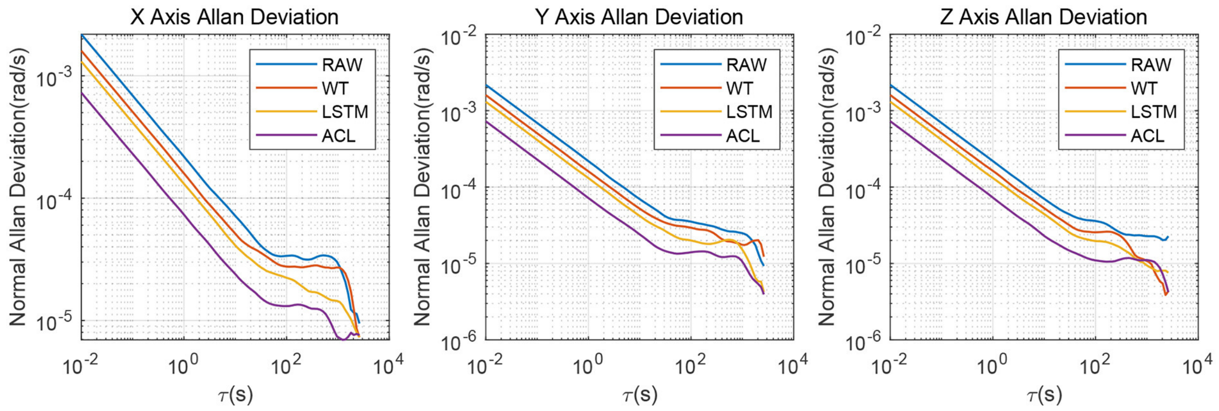


FIGURE 5
Allan variance comparison between denoised and raw signals for the 3-axis gyroscope of AHRS380SA.

TABLE 3 Allan variance parameters of the AHRS380SA 3-axis gyroscope.

Error sources	X-axis				Y-axis				Z-axis			
	Raw	WT	LSTM	ACL	Raw	WT	LSTM	ACL	Raw	WT	LSTM	ACL
Bias instability (deg/h)	9.77	8.76	5.31	4.07	8.14	6.05	5.59	3.76	8.06	6.96	5.29	3.31
Angle random walk ($deg/\sqrt{h}$)	0.75	0.55	0.45	0.25	0.75	0.55	0.45	0.25	0.74	0.56	0.45	0.26

the risk of overfitting and accelerate the training speed, the Adam optimizer (Kingma and Ba, 2014) with cosines warning restart scheduler (Loshchilov and Hutter, 2016) is adopted. To achieve the best performance of the ACL in the experiment, the parameters of the model are fully tuned. The hyperparameters are list in Table 2, where the learning rate is initialized at 0.0001, the batch size is set at 64, the dropout is 0.2, the number of CNN layer is 1, the size of CNN output channel is 256, the number of LSTM layer is 1, the size of LSTM output channel is 128, and 150 epochs of training are performed.

As illustrated in Figure 4, the X/Y/Z-axis raw data and the denoised data of WT, LSTM, and ACL methods are compared in blue, cyan, red, and green curves. LSTM and ACL can achieve significant noise reduction results for static signals better than traditional WT method, and the proposed ACL method has a better denoising effect than LSTM. The root means square error of them are 0.0022, 0.0016, 0.0013, and 0.00073 rad/s, respectively. Further, the Allan Variance curves comparison results are presented in Figure 5 and the specific error parameters are summarized in Table 3 to distinguish the differences between them. The results show that the ACL method performs the best noise reduction. Especially, the X-axis gyroscope has an improvement of 45.6 and 40.0% in bias instability and angle random walk using the LSTM neural network, while 58.3 and 66.6% using the ACL model. For the Y-axis gyroscope, the bias instability and angle random walk have a 31.3 and 40.0% improvement by the LSTM method, and 53.8 and 66.6% with the ACL model, respectively. For the Z-axis gyroscope, the error of bias instability and angle random walk are decreased by 34.4 and 39.2% using the LSTM method; meanwhile, the ACL model with 58.9 and 64.9%. Thus, according to the analysis of the static experiment, the proposed ACL method has good capability to restrain the stochastic error of the low-cost MEMS gyroscope compared with the application of the WT and LSTM neural network.

We further test the denoising performances of the proposed method in the dynamic condition. The AHRS380SA IMU is fixed on a turntable, the three axes of which are aligned with the three axes of the turntable. We set the turntable around the Z-axis as the Equation (11), and the sampling frequency is 100 Hz.

$$\omega = 2 * \sin(\pi t/500) \tag{11}$$

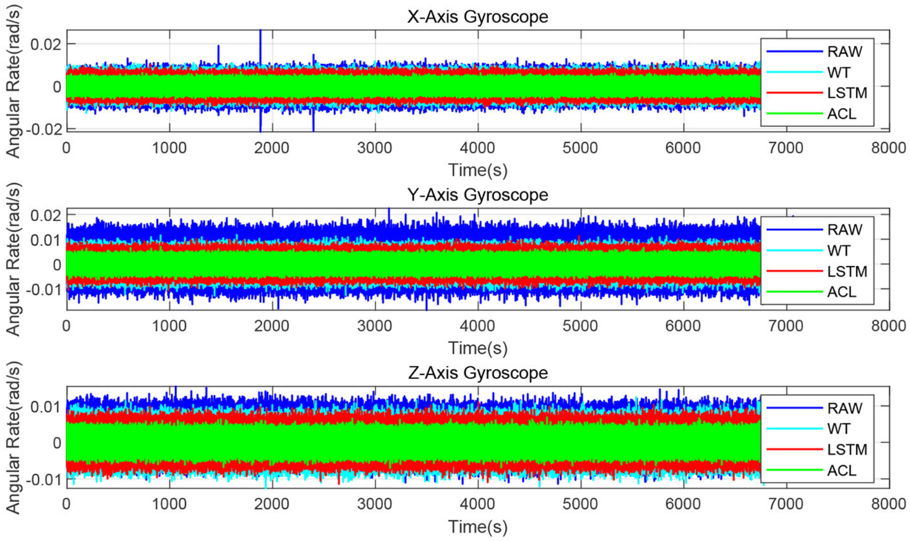


FIGURE 7
The denoised and raw signals comparison for the 3-axis gyroscope of XSENS MTI-G-700.

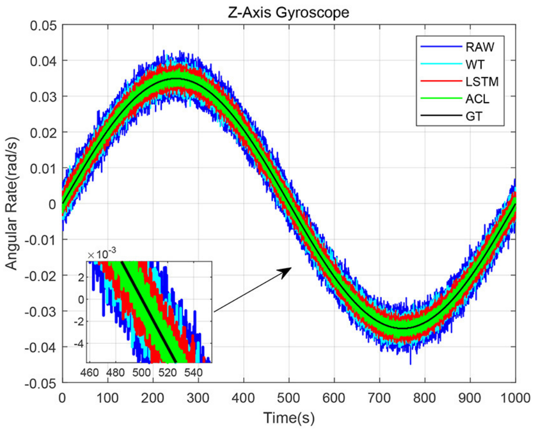


FIGURE 6
The results of dynamic tests for the AHRS380SA.

where ω is the angular velocity of the turntable.

Since Allan variance is generally used for static gyroscope data error analysis, root means square error (RMSE) is adopted as an accuracy evaluation index in dynamic experiments, which can reflect the distance between the denoised values and the actual ones. The smaller RMSE, the better the denoising effect, calculated as,

$$RMSE(\hat{y}, y) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{12}$$

where y is the actual value and $\hat{y}$ is the denoised value.

Figure 6 shows that three solutions have different effects on the z-axis gyroscope dynamic results. It can be seen that when the turntable angular velocity changes according to our previous setting value, the three denoising methods can track it well. Significantly, the ACL curve in green is closer to the ground truth (GT) curve in black than the LSTM curve in red by magnifying the period from 460 to 540 s. The raw data has the largest RMSE, i.e., 0.0022 rad/s. The WT and LSTM methods are better than the raw data results, and the RMSE are 0.0016 and 0.0013 rad/s, respectively. The ACL model has the best performance, the RMSE of which is 0.0007 rad/s.

4.2. Dataset tests

In order to further validate the proposed method, we conducted an open dataset test. Three MEMS IMU datasets with different accuracy in the famous kalibr—allan toolbox (Kalibr-Allan, 2017) are provided by the University of Delaware, i.e., XSENS MTI-G-700, Tango Yellowstone Tablet and ASL-ETH VI-Sensor.

Since the XSENS MTI-G-700 is a classic low-cost MEMS IMU in inertial navigation, we chose it as our test IMU. The XSENS MTI-G-700 dataset is continuously collected for 3 h at 400 Hz. Similar to real field tests, the results of raw, WT, LSTM and ACL methods are compared in Figure 7. These three noise reduction methods can reduce the peak and peak value of raw data to a certain extent, among which ACL is the best, LSTM is the second, and WT is the worst. The Allan variance curves comparison results are also depicted in Figure 8 and summarized in Table 4: (1) the raw data have the largest average

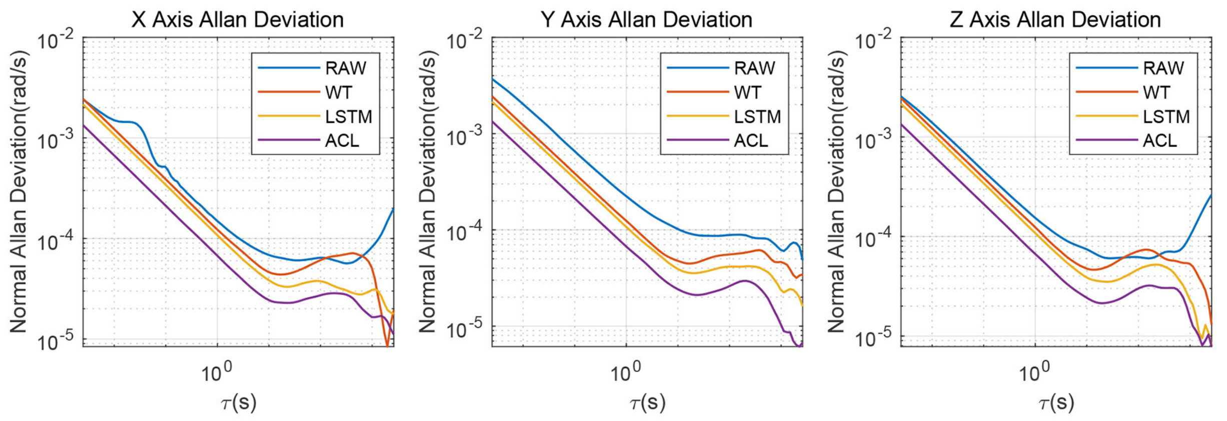


FIGURE 8
Allan variance comparison between denoised and raw signals for the 3-axis gyroscope of XSENS MTI-G-700.

TABLE 4 Allan variance parameters of the XSENS MTI-G-700 3-axis gyroscope.

Error sources	X-axis				Y-axis				Z-axis			
	Raw	WT	LSTM	ACL	Raw	WT	LSTM	ACL	Raw	WT	LSTM	ACL
Bias instability (deg/h)	20.11	13.62	11.73	7.10	25.30	14.92	12.89	8.14	21.9	13.39	10.90	7.23
Angle random walk $(deg/\sqrt{h})$	0.51	0.42	0.37	0.23	0.71	0.43	0.38	0.29	0.49	0.42	0.37	0.26

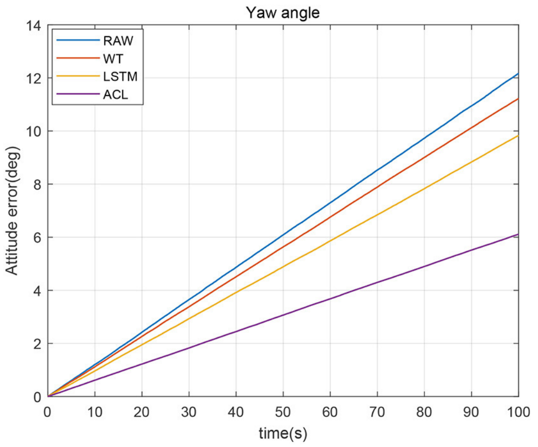


FIGURE 9
The yaw angle error of the denoised and raw signals.

bias instability, i.e., 22.4 deg/h; (2) the WT is better than the raw data, and the bias instability is reduced to 13.98 rad/s; (3) the LSTM has an 11.84 rad/s bias instability; (4) the ACL method performs best in the three solutions.

In order to further analyze the influence of stochastic error on the inertial navigation, we compared the denoised yaw angle errors in Figure 9. The yaw angle error gradually increased with time, and the maximum accumulation error reached 12.2 degrees after 100 s. If the error is not corrected, such a large yaw error cannot be used for inertial navigation. Compared with the yaw angle error of raw data, the denoised yaw angle error divergence over time is effectively improved, where the ACL method basically controls the maximum yaw accumulation error within 6 degrees. The WT and LSTM methods also reduce the degree of yaw angle divergence.

4.3. Ablation study

We evaluate the stochastic noise eliminating performance of removing the 1D-CNN and attention mechanism from the ACL model to demonstrate the effectiveness of the proposed ACL design choice of the 1D-CNN and attention mechanism. The ablation study is consist of the AHRS380SA denoised

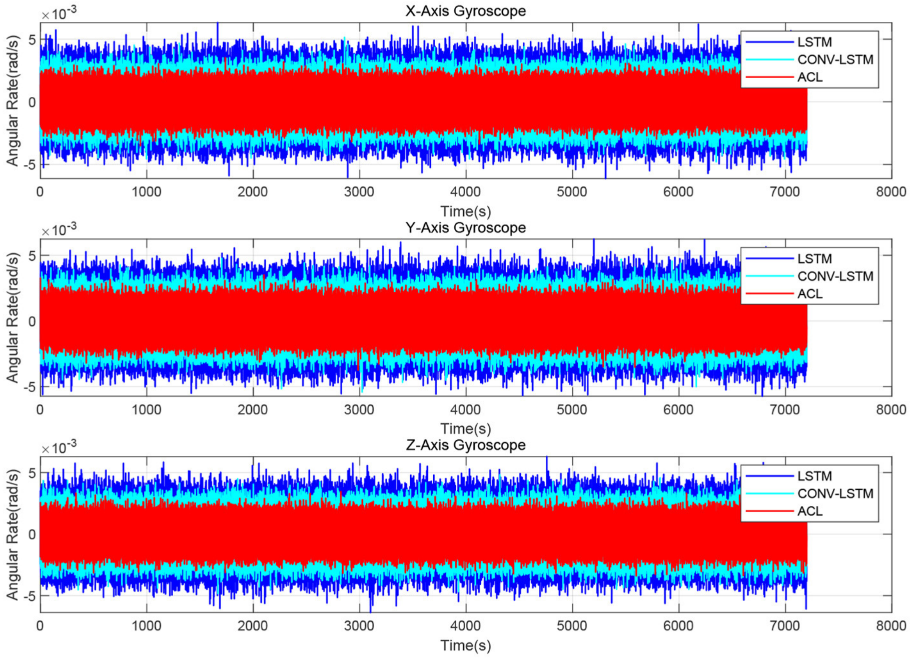


FIGURE 10
The denoised and raw signals comparison for the ablation study.

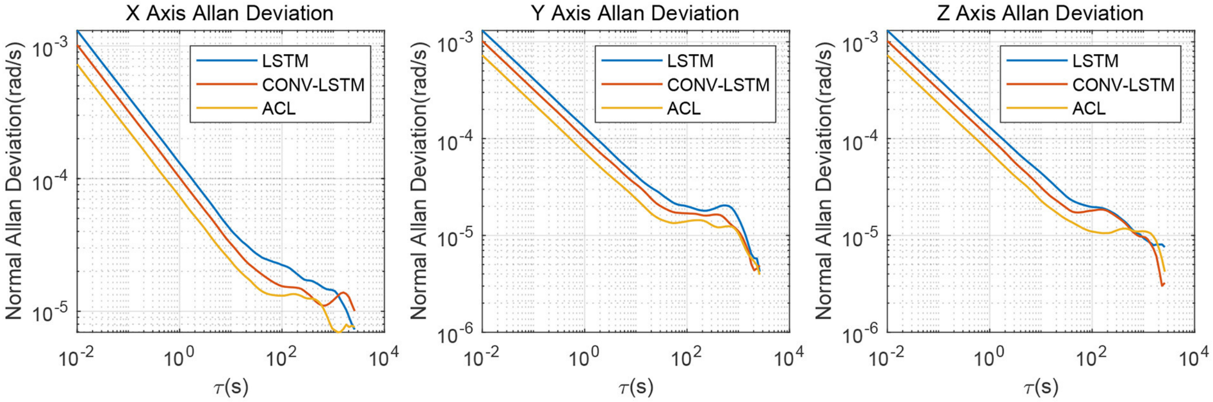


FIGURE 11
Allan variance comparison between denoised and raw signals for the ablation study.

performance comparison between the LSTM, CONV-LSTM, and ACL methods.

The denoised results for the AHRS380SA of all the ablation experiments are shown in Figures 10, 11. The average bias instability of the LSTM, CONV-LSTM and ACL are 5.40, 4.90, and 3.71 deg/h, meanwhile, the average angle random walks are 0.45, 0.32, and 0.24 $deg/\sqrt{h}$, respectively. Applying 1D CNN and attention mechanism in the ACL model has a lower stochastic error than without any attention mechanism after the LSTM layers. The ablation experiments show that all components in the ACL are effective.

5. Conclusion

This paper proposes a hybrid denoising method based on deep learning to reduce stochastic errors. The devised deep neural network architecture can predict the gyroscope measurements from various noises. Furthermore, the model combines 1D-CNN and LSTM to extract the local feature representation from the input multivariable time sequences and uses LSTM to correlate the current inputs and historical model information automatically. The attention mechanism is exploited to calculate the weight to improve computing efficiency. In order to verify the performance of the proposed method, numerical real field, dataset and ablation experiments have been performed. Comparing our algorithm with known work in this field, the evaluation results show that our model has greater denoising performances. However, there is still room for improvement, and further research can focus on improving the real-time capability. Furthermore, optimal deep learning based approaches (Reddy et al., 2018) and quantum recurrent network (Gandhi et al., 2013) will be explored for denoising gyroscope in future.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

YL designed the experiments and algorithms and wrote the main draft of the manuscript. JC and WL participated in the discussion and guided the paper writing. All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. doi: 10.48550/arXiv.1409.0473

Bhatt, D., Aggarwal, P., Bhattacharya, P., and Devabhaktuni, V. (2012). An enhanced MEMS error modeling approach based on Nu-support vector regression. *Sensors* 12, 9448–9466. doi: 10.3390/s120709448

Brossard, M., Bonnabel, S., and Barrau, A. (2020). Denoising IMU gyroscopes with deep learning for open-loop attitude estimation. *IEEE Robot. Automat. Lett.* 5, 4796–4803. doi: 10.1109/LRA.2020.3003256

Cai, S., Hu, Y., Ding, H., and Chen, H. (2018). A noise reduction method for is gyroscope based on direct modeling and Kalman filter. *IFAC Pap. Online* 51, 172–176. doi: 10.1016/j.ifacol.2018.10.032

Chen, C., Lu, X., Markham, A., and Trigoni, N. (2018). "IONet: learning to cure the curse of drift in inertial odometry," in *Proceedings of the AAAI Conference on Artificial Intelligence*. (New Orleans, LA). doi: 10.1609/aaai.v32i1.12102

Ding, M., Shi, Z., Du, B., Wang, H., and Han, L. (2021). A signal de-noising method for a MEMS gyroscope based on improved VMD-WTD. *Meas. Sci. Technol.* 32, 095112. doi: 10.1088/1361-6501/abfe33

El-Sheimy, N., Hou, H., and Niu, X. (2007). Analysis and modeling of inertial sensors using Allan variance. *IEEE Trans. Instrument. Meas.* 57, 140–149. doi: 10.1109/TIM.2007.908635

Esfahani, M. A., Wang, H., Wu, K., and Yuan, S. (2019). ORINet: robust 3-D orientation estimation with a single particular IMU. *IEEE Robot. Automat. Lett.* 5, 399–406. doi: 10.1109/LRA.2019.2959507

Gandhi, V., Prasad, G., Coyle, D., Behera, L., and McGinnity, T. M. (2013). Quantum neural network-based EEG filtering for a brain-computer interface. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 278–288. doi: 10.1109/TNNLS.2013.2274436

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to forget: continual prediction with lstm. *Neural Comput.* 12, 2451–2471. doi: 10.1162/089976600300015015

Gonzalez, R., and Catania, C. A. (2019). Time-delayed multiple linear regression for de-noising MEMS inertial sensors. *Comput. Electric. Eng.* 76, 1–12. doi: 10.1016/j.compeleceng.2019.02.023

Graves, A., Fernández, S., and Schmidhuber, J. (2005). "Bidirectional LSTM networks for improved phoneme classification and recognition," in *International Conference on Artificial Neural Networks* (Warsaw: Springer), 799–804. doi: 10.1007/11550907_126

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2020). A survey on visual transformer. *arXiv preprint arXiv:2012.12556*. doi: 10.48550/arXiv.2012.12556

Han, S., Meng, Z., Zhang, X., and Yan, Y. (2021). Hybrid deep recurrent neural networks for noise reduction of MEMS-IMU with static and dynamic conditions. *Micromachines* 12, 214. doi: 10.3390/mi12020214

Jiang, C., Chen, S., Chen, Y., Bo, Y., Han, L., Guo, J., et al. (2018a). Performance analysis of a deep simple recurrent unit recurrent neural network (SRU-RNN) in MEMS gyroscope de-noising. *Sensors* 18, 4471. doi: 10.3390/s18124471

Jiang, C., Chen, S., Chen, Y., Zhang, B., Feng, Z., Zhou, H., et al. (2018b). A MEMS IMU de-noising method using long short term memory recurrent neural networks (LSTM-RNN). *Sensors* 18, 3470. doi: 10.3390/s18103470

Kalibr-Allan (2017). Kalibar Allan toolbox. Available online at: https://github.com/rpng/kalibr_allan

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/arXiv.1412.6980

Koroteev, M. (2021). Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*. doi: 10.48550/arXiv.2103.11943

Leung, H., Lo, T., and Wang, S. (2001). Prediction of noisy chaotic time series using an optimal radial basis function neural network. *IEEE Trans. Neural Netw.* 12, 1163–1172. doi: 10.1109/72.950144

Li, D., Zhou, J., and Liu, Y. (2021). Recurrent-neural-network-based unscented kalman filter for estimating and compensating the random drift of MEMS gyroscopes in real time. *Mech. Syst. Signal Process.* 147, 107057. doi: 10.1016/j.ymssp.2020.107057

Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., et al. (2020). TLIO: tight learned inertial odometry. *IEEE Robot. Automat. Lett.* 5, 5653–5660. doi: 10.1109/LRA.2020.3007421

Loshchilov, I., and Hutter, F. (2016). SGDR: stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*. doi: 10.48550/arXiv.1608.03983

Reddy, T. K., Arora, V., and Behera, L. (2018). HJB-equation-based optimal learning scheme for neural networks with applications in brain-computer interface. *IEEE Trans. Emerg. Top. Comput. Intell.* 4, 159–170. doi: 10.1109/TETCI.2018.2858761

Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*. doi: 10.21437/Interspeech.2014-80

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenomena* 404, 132306. doi: 10.1016/j.physd.2019.132306

Shiau, J. K., Ma, D. M., Huang, C. X., and Chang, M. Y. (2011). "MEMS gyroscope null drift and compensation based on neural network," in *Advanced Materials Research*, Vol. 255 (Trans Tech Publications), 2077–2081. doi: 10.4028/www.scientific.net/AMR.255-260.2077

Song, J., Shi, Z., Wang, L., and Wang, H. (2018). Improved virtual gyroscope technology based on the arma model. *Micromachines* 9, 348. doi: 10.3390/mi9070348

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*. eds. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Q. Weinberger (Montreal, QC: Neural Information Processing Systems).

Woodman, O. J. (2007). *An Introduction to Inertial Navigation*. Technical report, University of Cambridge, Computer Laboratory.

Yuan, J., Yuan, Y., Liu, F., Pang, Y., and Lin, J. (2015). An improved noise reduction algorithm based on wavelet transformation for MEMS gyroscope. *Front. Optoelectron*. 8, 413–418. doi: 10.1007/s12200-015-0474-2

Zhang, Q., Wang, X., Wang, S., and Pei, C. (2018). Application of improved fast dynamic allan variance for the characterization of MEMS gyroscope on UAV. *J. Sens*. 2018, 2895187. doi: 10.1155/2018/2895187

Zhang, S., Yu, S., Liu, C., Yuan, X., and Liu, S. (2016). A dual-linear Kalman filter for real-time orientation determination system using low-cost MEMS sensors. *Sensors* 16, 264. doi: 10.3390/s16020264

Zhang, Y.-S., and Yang, T. (2012). Modeling and compensation of MEMS gyroscope output data based on support vector machine. *Measurement* 45, 922–926. doi: 10.1016/j.measurement.2012.02.001

Zhao, H., Jia, J., and Koltun, V. (2020). "Exploring self-attention for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA), 10076–10085. doi: 10.1109/CVPR42600.2020.01009

Zhu, C., Cai, S., Yang, Y., Xu, W., Shen, H., and Chu, H. (2021). A combined method for MEMS gyroscope error compensation using a long short-term memory network and Kalman filter in random vibration environments. *Sensors* 21, 1181. doi: 10.3390/s21041181

TYPE Original Research
PUBLISHED 09 January 2023
DOI 10.3389/fnbot.2022.1053067

OPEN ACCESS

EDITED BY
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

REVIEWED BY
Honglei Xu,
Curtin University, Australia
Jun Wang,
Western Sydney University, Australia
Jiaxi Wang,
Zhejiang Lab, China

*CORRESPONDENCE
Jianjun He
✉ jjhe@csu.edu.cn
Shuai Shen
✉ shenshuaiokay@163.com

RECEIVED 25 September 2022
ACCEPTED 14 December 2022
PUBLISHED 09 January 2023

CITATION
Hu E, He J and Shen S (2023) A dynamic integrated scheduling method based on hierarchical planning for heterogeneous AGV fleets in warehouses.
Front. Neurorobot. 16:1053067.
doi: 10.3389/fnbot.2022.1053067

A dynamic integrated scheduling method based on hierarchical planning for heterogeneous AGV fleets in warehouses

Enze Hu, Jianjun He* and Shuai Shen*

The School of Automation, Central South University, Changsha, China

In modern industrial warehouses, heterogeneous and flexible fleets of automated guided vehicles (AGVs) are widely used to improve transport efficiency. However, as their scale and limit of battery capacity increase, the complexity of dynamic scheduling also increases dramatically. The problem is to assign tasks and determine detailed paths to AGVs to keep the multi-AGV system running efficiently and sustainedly. In this context, a mixed-integer linear programming (MILP) model is formulated. A hierarchical planning method is used, which decomposes the integrated problem into two levels: the upper-level task-assignment problem and the lower-level path-planning problem. A hybrid discrete state transition algorithm (HDSTA) based on an elite solution set and the Tabu List method is proposed to solve the dynamic scheduling problem to minimize the sum of the costs of requests and the tardiness costs of conflicts for the overall system. The efficacy of our method is investigated by computational experiments using real-world data.

KEYWORDS

automated guided vehicles, dynamic integrated scheduling, task assignment, path planning, hierarchical planning, hybrid discrete state transition algorithm

1. Introduction

With the development in automation technology, AGVs as an important component of the modern warehouse logistics system is getting increased attention because of their accuracy, flexibility, and efficiency. More recently, heterogeneous AGV fleets are rapidly being adopted by industrial instances to perform different material handling tasks, where each vehicle has specific capabilities (e.g., pallet truck AGVs can tow loads, while backpack AGVs can lift loads). The minimization of travel costs is the most important objective of dynamic scheduling pursued in practice, which is affected by various decisions such as task assignment (i.e., assigning and sequencing tasks to AGVs), path planning (i.e., selecting optimal paths taken by each vehicle to reach the destination), and conflict management (i.e., avoiding conflicts between AGVs). These subproblems are interdependent; therefore, optimizing scheduling problems sequentially may yield a suboptimal performance of the overall AGV system (Maza and Castagna, 2005).

An example of a warehouse trying to implement an automated material handling system using a heterogeneous AGV fleet is Trucking Company (TC), which is a high-tech listed company in Changsha, China. Currently, the vehicle management system used in TC relies on prepackaged software provided by AGV manufacturers. However, such software packages are not applicable to a heterogeneous AGV fleet and cannot handle dynamic problems such as the addition of new tasks and charging requests for AGVs. In addition, the optimal task assignment scheme may cause more traffic jams during path planning, and an evaluation index needs to be quantified and established for the delay time caused as a result of the waiting and detour strategy of the AGVs. As the configuration cost increases, a real-time and efficient integrated scheduling method becomes important in improving the economic performance of the warehouse. In this study, we focus on a dynamic integrated scheduling problem for heterogeneous AGVs with battery constraints.

Motivated by our collaboration with the TC, the main novelty of our problem setting in contrast to the existing literature is constituted by the combination of the following features. First, we specifically focused on solving the scheduling problems right on time, whereas the methods in most studies consume unreasonable computational effort, in particular, some exact methods (Schiffer and Walther, 2017; Ma et al., 2020; Singh et al., 2022). Second, the AGVs considered in this study are heterogeneous in terms of battery management, travel speed, and capabilities to perform transportation of different types of materials, which increases the complexity of the problem. Third, we simultaneously considered joint task assignments, path planning, and conflicts that reduce the problems of the AGV system. We aimed to make decisions on optimizing the overall AGV system performance rather than successively solving each subproblem. Our main contributions are summarized as follows:

First, we developed a mixed-integer linear programming (MILP) model for analyzing the scheduling of multi-AGVs, which combines both task assignment and path planning in automated warehouses. The model captures conflicts between a heterogeneous set of AGV fleets, allowing for scheduling according to the uncertain environment. The objective is to minimize the sum of the costs of requests and costs of conflicts. Constraints are also formulated to cope with features of capacity and battery management.

Second, the hierarchical planning method was used to decompose the complex and integrated scheduling problem. We propose a hybrid discrete state transition algorithm (HDSTA) considering the two-layer problems based on incorporating an elite solution set and the Tabu Search to find the optimal solution for the overall system instead of optimal solutions for each independent problem. Although our model is stylized for warehouses, the method can be applied to other applications such as flexible manufacturing systems and automated container terminals.

Third, we present the concept of a path expert database and its generation methods. The selection procedure based on a preset database is established for real-time path planning, which provides the foundation for dynamic scheduling.

Finally, numerical experiments are performed to validate the model according to the real-world data of warehouses in Changsha, China. Our approach is shown to yield approximate optimal solutions for AGV scheduling and path planning within a reasonable timeframe.

The remainder of this article is organized as follows. Relevant literature on the scheduling of multi-AGVs is discussed in the "Literature review" section. Problem description and the MILP model are formally established in the "Dynamic scheduling system and problem description" section. The "Hierarchical planning method" section presents the hierarchical planning method and introduces the proposed HDSTA and the selection procedure based on the path expert database. The "Computational experiments" section reports the experiments conducted to test the proposed method. Finally, conclusions and several future research directions are discussed in the "Conclusion" section.

2. Literature review

AGV scheduling can directly determine the efficiency and the cost of the overall transport system and therefore high attention is paid by researchers or manufacturing enterprises. Fazlollahtabar and Saidi-Mehrabad (2015) presented a literature review and divided AGV scheduling into three subproblems, task assignment, path planning, and collision avoidance. Many studies applied various methods, such as exact methods, heuristics, and meta-heuristics, to treat the subproblems separately or simultaneously.

As for exact methods, Desaulniers et al. (2003) designed an exact method including three algorithms (greedy search, column generation, and branch cutting), which enables solving the scheduling problem for four vehicles. Nishi et al. (2011) addressed a Lagrangian relaxation and cut scheme under the bilevel decomposition framework to optimize simultaneous task assignments and conflict-free routing problems. Fazlollahtabar and Hassanli (2018) presented a modified network simplex algorithm for blocking a scheduling problem in the manufacturing system. Nevertheless, because of the non-deterministic polynomial-time (NP)-hard nature of the scheduling problems, the exact method is only suitable for instances of small-scale problems.

For large-scale complex real-world problems, heuristics or metaheuristics are mainly adopted. Li et al. (2019) proposed an improved harmony search algorithm to improve the AGV scheduling rate, which can obtain the best harmony by considering the rate change. Zhang et al. (2019) proposed a genetic algorithm and a hybrid-load AGV scheduling model to reduce the total cost of the logistics system, which was successfully applied to a mixed-model automobile assembly line. Abderrahim et al. (2020) used a variable neighborhood search algorithm to assign tasks in a manufacturing shop based on a vehicle manufacturing facility to minimize the maximum completion time. Zhang et al. (2022) proposed an improved iterated greedy algorithm to solve the AGV dispatching problem to minimize the total transportation cost of the matrix manufacturing workshop. In addition, many other meta-heuristics were also used in scheduling problems, such as the simulated annealing algorithm (Lu and Wang, 2019), the two-stage ant colony algorithm (Hamzeei et al., 2013), the evolutionary algorithm (Saidi-Mehrabad et al., 2015), and the particle swarm optimization algorithm (Gen et al., 2017). In the above literature, a common feature of the problems studied is that all the task information are stable and obtained in advance, and then, an analytical model was established and the problems are solved with a heuristic or meta-heuristic algorithm. Nevertheless, in a real-world instance, it is unrealistic to obtain all the task information in advance, while many uncertainties (e.g., urgent tasks and task rework) exist under dynamic and complex environments (Zhang et al., 2017). Therefore, the static scheduling method is insufficient for the complicated real-world industrial environment.

In recent years, with the development of IoT technology, many researchers focused on the dynamic scheduling problem. Li et al. (2020) proposed a multi-vehicle AGV scheduling mechanism for simulating multicustomer demands in an intelligent warehouse system. Mourtzis et al. introduced a cloud-based cyber-physical system with the help of IoT to achieve adaptive shop floor scheduling and condition-based maintenance. Umar et al. (2015) proposed an improved hybrid genetic algorithm method for dynamic scheduling that considers dispatching and conflict-free routing problems of AGVs under a flexible workshop environment. Lyu et al. (2019) presented an improved genetic algorithm combined with the Dijkstra algorithm considering time windows to solve the problems of optimal numbers, shortest transportation time, and conflict-free routing in the path planning process. Qiuyun et al. (2021) improved the particle swarm optimization algorithm to obtain the shortest transportation time for the AGV path planning problem of a one-line production line in manufacturing. Guo et al. (2020) studied the acceleration control method and the AGV priority determination method to improve the negotiation of AGVs that implement conflict-free path planning. Nevertheless, these researchers ignored the influence of not only the case of AGV heterogeneity but also battery management.

Through the review of the above literature, there have been no studies on dynamic integrated scheduling in warehouses for a heterogeneous set of AGV fleets with battery constraints. Therefore, a novel scheduling approach for AGVs is in high demand. In this study, we propose an HDSTA under a hierarchical planning framework to solve the complex problem, which is a kind of intelligent optimization algorithm with good global search capability and convergence property, considering the solution as a state and the update of the solution as a state transition process. Thus, we evaluated the proposed method with an industrial case study finally.

3. Dynamic scheduling system and problem description

In this section, a dynamic scheduling system for AGVs is proposed, which is based on a control system using inertial navigation guidance and QR codes. The information service is implemented by network and wireless routers. The integrated scheduling problems of heterogeneous AGVs with battery constraints in the AGV system are described and formulated while the conflict problem is highlighted.

3.1. Dynamic schedule system

The overall architecture of the dynamic scheduling system is presented in Figure 1. The dynamic supervisory layer provides real-time information about AGVs and the current schedule. The AGV monitoring system is responsible for managing the AGVs in terms of recognition, positioning, motor control, and battery level. The schedule monitoring system is responsible for receiving new tasks while monitoring the implementation of the current schedule and requesting a new schedule as a result of a change of tasks. The rescheduled module initialization harmonizes additional parameters with the running schedule that includes active AGVs, new tasks, completed tasks, and in-process tasks.

The integrated scheduling layer is responsible for determining the task assignment/sequence and path planning, which is more complex because of the consideration of conflict avoidance. AGV movement on warehouse layout is a multigraph problem, in that there are various parallel paths between the presorting stations. The path expert database is established in the offline stage, which can be regarded as a dataset containing warehouse layout information and the candidate elite paths sets between each presorting station. Accessory equipment such as sensors are equipped which enables AGVs to detect moving objects by hardware and avoid collision by preprocessed combination strategies of traffic regulations (e.g., stop and wait for the higher priority AGV to pass first or move around the conflict location). Conflicts can also be reduced by combining and changing the task assignment/sequence if it cannot be solved separately by path selection. However, the delay time as a result of the waiting and detour strategy of AGVs needs to be quantified and reduced. The schedule contains task assignment/sequence and path planning, which are generated by the task scheme generator and the path planning generator. The generated schedule is downloaded for execution by the system.

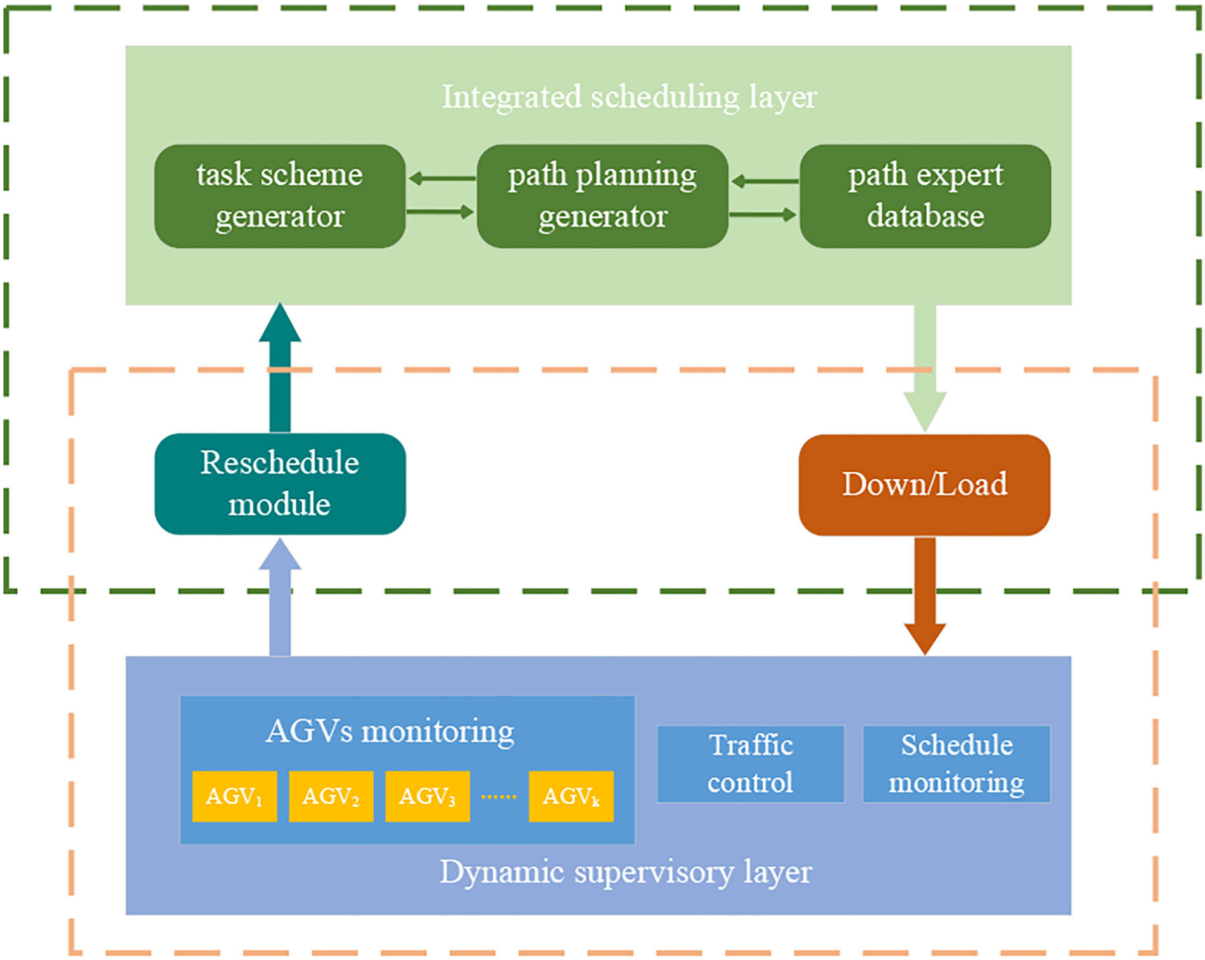


FIGURE 1
Dynamic scheduling system.

In a dynamic system, the assumptions considered are as follows: (1) loading and unloading times are fixed; (2) the AGVs move in four directions; and (3) the positioning deviation of the AGVs is negligible.

3.2. Problem statement

This study considers a real-world industry case of the TC where the goal is to have continuous material handling without human interference. Each transport process of the AGVs is composed of pickup travel, loading, delivery travel, and unloading. The layout of the warehouse is modeled as a multigraph, $G = (N, E)$, where $N = \{1, 2, \dots, n\}$ is a set of all the nodes. Let $C \subset N$ denote the set of charging stations and $X \subset N$ denote the set of presorting stations, respectively. Moreover, $E = \{(i, j, p) : i, j \in N, i \neq j\}$ denotes the set of arcs between every node pair. The p^{th} path between the nodes i and j is represented by $(i, j, p) \in E$. Parallel paths are stored in the path expert database N_s which is established in the offline stage. If there is a collision between a pair of current paths, a path parallel to one of this pair of current paths can be used to replace this path for avoiding conflict.

In our problem setting, a set of transport tasks T are serviced by a set of heterogeneous AGVs K, and each task $r \in T$ contains a pickup node and a delivery node which are denoted by $u_r \in X$ and $d_r \in X$, respectively. Besides transport task requests, a set of charging requests is denoted by $B = \{1, 2, \dots, |B|\}$, where $|B| = |C| \bullet |K|$ is the upper bound which is sufficiently large and C represents a set of charging stations. For each charging request $b \in B$, the pickup node and the delivery node are the same. The AGV makes a start instruction at the origin station s and each request contains only one delivery node. A termination instruction will be issued when the AGV reaches the terminal station e. Multiple request sets are defined by $R = T \cup B, R_s = R \cup \{s\}, R_e = R \cup \{e\}$, and $R_{se} = R \cup \{s\} \cup \{ e\}$.

We considered battery constraints and the maximum and minimum battery levels for each AGV, where $k \in K$ are denoted by b_h^k and b_l^k, respectively. Before performing a new task, the battery level b_k of each AGV needs to be above the minimum threshold b_l^k. The AGV is not allowed to access the charging facilities while traveling with a load. AGVs are required to complete the current task with the first priority before going to the charging station. The discharging rate of AGV $k \in K$ is represented by d_k, while each AGV has a unit time travel cost of c_k.

As previously mentioned, AGVs are heterogeneous in terms of their capabilities to perform the transportation of different types of materials. Let C_r^T denote a set of capability requirements for each task, and each AGV has a specific capability C_k^K. The task $r \in T$ is only able to be performed by AGV $k \in K$ if $C_r^T \subseteq C_k^K$ holds to ensure that each task is performed by an AGV with corresponding handling capacity. For example, the tasks of lifting loads need to be performed by backpack AGVs while the pallet truck AGVs are only able to tow loads.

The path from the pickup node to the delivery node r is denoted by P_r, and the path from the delivery node of task r to the pickup node of task r' is denoted by $P_{rr'}$. The AGV has its specific forward speed and velocity of rotation, respectively. For each path $p \in P_r$, the travel time of AGV k is denoted by T_{rp}^k, and for each path $p \in P_{rr'}$, the travel time of AGV k is denoted by $T_{rr'p}^k$. In addition, the conflict-free path is optional during path planning because collisions can be prevented by traffic regulations. A delay time returns when an AGV follows a waiting and detour strategy to avoid a collision. When the path $p \in P_r$ of AGV k conflicts with the path $q \in P_m$ of AGV g, the delay time of AGV k on the path $p \in P_r$ is defined by $\Phi_{rkp}^{mgq}(z_{rk}^u, z_{mg}^u)$, where z_{rk}^u represents the time for AGV k to arrive at the pickup node of request r and z_{mg}^u represents the time for AGV g to arrive at the pickup node of request m, respectively. When the path $p \in P_r$ of AGV k conflicts with the path $q \in P_{mm'}$ of AGV g, the delay time of AGV k on the path $p \in P_r$ is defined by $\Phi_{rkp}^{mm'gq}(z_{rk}^u, z_{mg}^d)$, where z_{mg}^d represents the time for AGV g to arrive at the delivery node of request m.

3.3. Mixed-integer linear programming model

In this section, we formulate a mathematical model based on the problem description, which is an improvement from the findings of Dang et al. (2021) and Singh et al. (2022). Decision variables are introduced as follows:

x_{rk}^p: binary variable equal to 1 if AGV $k \in K$ travels from the pickup node to the delivery node of request $r \in R$ using $p \in P_r$ or 0 otherwise

$y_{rr'k}^p$: binary variable equal to 1 if AGV $k \in K$ travels from the delivery node of request $r \in R_s$ to the pickup node of request $r' \in R_e$ using $p \in P_{rr'}$ or 0 otherwise

z_{rk}^u: time of AGV k at the pickup node u_r of request $r \in R_e$; z_{ek}^u is the termination time of AGV k

z_{rk}^d: time of AGV k at the delivery node d_r of request $r \in R_s$; z_{sk}^d is the start time of AGV k

λ_{rk}^u: percent amount of battery discharge of AGV k at the pickup node u_r of request $r \in R_e$

λ_{rk}^d: percent amount of battery discharge of AGV k at the delivery node d_r of request $r \in R_s$

The mathematical model of the described problem is presented as follows:

The objective function f is to minimize the sum of the costs of requests and the tardiness costs of conflicts as the cost of each AGV is directly proportional to its travel time.

$$f = \min \sum_{k \in K} c_k (z_{ek}^u - z_{sk}^d)$$

$$\sum_{k \in K} \sum_{p \in p_r} \mathrm{x}^p{}_{rk} = 1 \quad \forall r \in T \tag{1}$$

$$\sum_{k \in K} \sum_{p \in p_r} \mathrm{x}^p{}_{rk} \leq 1 \quad \forall r \in B \tag{2}$$

$$x^p{}_{rk} = 0 \quad \forall p \in P_r,\ \forall r \in T, \forall k \in K, C_r^T \not\subseteq C_k^K \tag{3}$$

$$y^p{}_{rrk} = 0 \quad \forall r \in R, \forall k \in K \tag{4}$$

$$\sum_{r \in R_s} \sum_{p \in p_{rr'}} y^p{}_{rr'k} = \sum_{p \in p_{r'}} x^p{}_{r'k} \quad \forall r' \in R, \forall k \in K \tag{5}$$

$$\sum_{p \in p_r} x^p{}_{rk} = \sum_{r' \in R_e} \sum_{p \in p_{rr'}} y^p{}_{rr'k} \quad \forall r \in R, \forall k \in K \tag{6}$$

$$\begin{aligned} z_{rk}^u + T_{rk}^p &+ \sum_{g \in K} \sum_{m \in T} \sum_{q \in p_m} \Phi_{rkp}^{mgq} (z_{rk}^u, z_{mg}^u) x^q{}_{mg} \\ &+ \sum_{g \in K} \sum_{m \in R_s} \sum_{m' \in R_e} \sum_{q \in p_{mm'}} \Phi_{rkp}^{mm'gq} (z_{rk}^u, z_{mg}^d) \\ &* y^q{}_{mm'g} - M(1 - x^p{}_{rk}) \leq z_{rk}^d \quad \forall p \in P_r, \\ &\forall r \in T, \forall k \in K \end{aligned} \tag{7}$$

$$\begin{aligned} z_{rk}^d + T_{rr'k}^p &+ \sum_{g \in K} \sum_{m \in T} \sum_{q \in p_m} \Phi_{rr'kp}^{mgq} (z_{rk}^d, z_{mg}^u) x^q{}_{mg} \\ &+ \sum_{g \in K} \sum_{m \in R_s} \sum_{m' \in R_e} \sum_{q \in p_{mm'}} \Phi_{rr'kp}^{mm'gq} (z_{rk}^d, z_{mg}^d) \\ &* y^q{}_{mm'g} - M(1 - y^p{}_{rr'k}) \leq z_{r'k}^u \\ &\forall p \in P_{rr'}, \forall r \in R_s, \forall r' \in R_e, \forall k \in K \end{aligned} \tag{8}$$

Constraint (1) ensure that each transport request is assigned only one time and can be followed by another request. Constraint (2) makes sure that each charging request is presented by at most one AGV. Each charging request r has only one path. Constraint (3) ensures that the capabilities of the requests and AGVs match. Constraint (4) ensures that self-visits are avoided. Constraints (5) and (6) make sure that the number of entering paths of request, execution paths, and leaving paths of each request are consistent. The time constraints are given by (7) and (8), and Constraint (7) calculates the travel time from the pickup node of request r to the delivery node. Constraint (8) calculates the travel time between different requests. The travel time includes transportation time and delay time, where M is a large positive constant.

$$b_l^k \leq \lambda_{rk}^d \leq b_h^k \quad \forall r \in R_s, \forall k \in K \tag{9}$$

$$b_l^k \leq \lambda_{rk}^u \leq b_h^k \quad \forall r \in R_e, \forall k \in K \tag{10}$$

$$\lambda_{rk}^u + d_k(z_{rk}^d - z_{rk}^u) - M(1 - \mathrm{x}^p{}_{rk}) \leq \lambda_{rk}^d$$
$$\forall p \in P_r, \forall r \in T, \forall k \in K \tag{11}$$

$$\lambda_{rk}^d + d_k(z_{r'k}^u - z_{rk}^d) - M(1 - \mathrm{y}^p{}_{rr'k}) \ \leq \lambda_{r'k}^u \quad \forall p \in P_{rr'},$$
$$\forall r \in R_s, \forall r' \in R_e, \forall k \in K \tag{12}$$

The constraints related to power consumption are given by (9) to (12). Constraints (9) and (10) set the lower and upper bounds for an amount of battery discharge. Constraint (11) calculates the amount of battery discharge due to the travels between the source and destination of a request. Constraint (12) calculates the amount of battery discharge due to the travel between the destination and the source of two requests.

$$x^p{}_{rk} \in \{0, 1\} \quad \forall p \in P_r,\ \forall r \in R, \forall k \in K \tag{13}$$

$$y^p{}_{rr'k} \in \{0, 1\} \forall p \in P_{rr'} \quad \forall r \in R_s, \forall r' \in R_e, \forall k \in K \tag{14}$$

$$z_{rk}^u \geq 0 \quad \forall t \in R_e, \forall k \in K \tag{15}$$

$$z_{rk}^d \geq 0 \quad \forall t \in R_s, \forall k \in K \tag{16}$$

The valid domains of the binary variables are given by constraints (13)–(16), which guarantee valid domains for the other decision variables.

4. Hierarchical planning method

In this section, a hierarchical planning method is proposed to solve the joint task assignments, path planning, and conflict problem for just-in-time scheduling. This method is inspired by the work of Hooker and Ottosson (2003) and decomposes the integrated optimization problems into an aggregated upper-level master problem and a lower-level subproblem. The upper-level problem is to make decisions for AGV task assignment/sequence, which determines a candidate elite solution set where the collision constraints for AGVs are neglected. The lower-level subproblem is to solve the optimal path planning problem with collision constraints under the conditions of the tentative solution at the upper level. The conflict problem is considered in both the master problem and subproblem, the collision between AGVs can be reduced by changing the detailed paths for vehicles or the scheme of task assignment and sequence. In summary, the objective is to minimize the AGV transportation time, which is the sum of the total travel time and the delay time (waiting or detour time for avoiding collisions). The detailed steps are described as follows:

Step 1. The upper level: Task assignment and sequence to AGVs where the collision constraints are removed from the original problem, and the transportation time of each task for each AGV is defined as the minimal time from the starting node to the delivery node. The master problem is regarded as the task assignment/sequence problem with constraints such as heterogeneous AGVs and batteries. In this study, a tentative elite solution set φ_i sorted in the ascending order of the objective function value is generated by HDSTA where the solution in the elite solution set is denoted by p_n.

Step 2. The lower level: Select the specific paths to perform the assigned tasks for the AGVs under the condition that a tentative solution p_n is derived from a master problem. For each solution, $p_n \in \varphi_i$ selected in the ascending order, the subproblem, which is concerned with the path planning problem to select the optimal paths with collision constraints for AGVs, is solved by the select procedure, while a list of conflict results with memories is generated, called Tabu List Λ_i. If the result of the selected procedure is conflict-free paths (termination criterion 1), the algorithm is completed; otherwise, recording conflict results to Λ_i. In the iteration, the solution with the minimum objective function values is recorded as the tentative optimal solution p_{best} and its delay time is defined by t_p.

Step 3. Algorithm termination criterion 2: The maximum allowable delay time is defined by ε. If t_p derived in Step 2 is less than ε, the algorithm is completed.

Step 4. Regenerate the tentative elite solution set φ_i considering the information is recorded in the tabu list Λ_i by HDSTA. If there is no improvement in the objective function value after five iterations, the algorithm is completed (termination criterion 3); otherwise, updating Λ_i and returning to Step 2.

The main scheme of hierarchical planning is illustrated in Algorithm 1. The accurate information about all AGVs and tasks are known, and the path expert database must be computed offline in advance.

An HDSTA with a path-select procedure and tabu list is proposed to find the optimal solution. The algorithm starts with a dynamic serve framework by generating a reschedule at the appropriate time interval methodology, based on the concept of dynamic scheduling, when there is a requirement for an additional task or AGV charging (lines 1–5). Then, the initialization of the tentative elite solution set φ_i using HDSTA (line 6) was carried out. For each solution $p_n \in \varphi_i$, detailed paths are generated using the path select procedure

```
Require: set of AGVs $K$, set of tasks $T$,
path expert database $N_s$
1: if there is a change in transport or
charging requires begin
2:   Extract finished tasks from the running
     schedule
3:   Combine the remaining tasks and new
     tasks
4:   Update $K$, $T$
5: end
6: Initialize tentative elite solution set
$\varphi_i$ by HDSTA
7: while ($y_1 = 1$ or $y_2 = 1$or $y_3 = 1$)
8:  for each $p_n \in \varphi_i$
9:   $[p, T, t_p, S_p] \leftarrow$ SP ($p_n$)
10:   if $t_p = 0$ then
11:    $D \leftarrow p_n$; $P_{best} \leftarrow p$; $T_{best} \leftarrow T$; $y_1 \leftarrow$ 1
12:   break for;
13:   end if
14:   if $t_p < t_{best}$ then
15:    $t_{best} \leftarrow t_p$; $P_{best} \leftarrow p$; $T_{list} \leftarrow [p_n, S_p]$; $D \leftarrow$
       $p_n$; $T_{best} \leftarrow T$
16:   end if
17:   end for
18:   if $t_{best} < \varepsilon$ then
19:    $y_2 \leftarrow$ 1
20:   end if
21:   if $y_1 = 0$ or $y_2 = 0$ then
22:     update the elite solution set $\varphi_i$ by
        HDSTA
23:     update $l$
24:   if $l > 5$
25:   $y_3 \leftarrow$ 1
26:   end if
27:   end if
28: end while
29: $S \leftarrow (D, P_{best})$
return $S$
```

Algorithm 1. Main scheme of hierarchical planning.

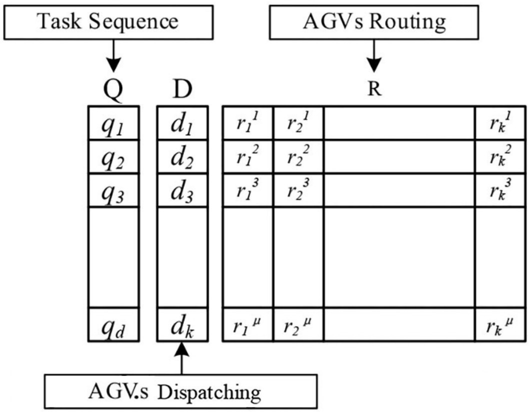


FIGURE 2
State space representation.

```
Require: path expert database $N_s$,
dispatching $D$
$T_c \leftarrow$ 0
repeat
for each $d_k \in D$
  for each $t \in T_k$
    $r_u^k \leftarrow Path(o_u^t, 1)$; $r_{u'}^k \leftarrow Path(o_d^t, 1)$
    $r_t \leftarrow r_u^k$; $r_t \leftarrow r_{u'}^k$; $R_k \leftarrow r_t$
  end for
  $R \leftarrow R_k$
end for
[$T_c$, $C_i$] $\leftarrow$ Con ($R$)
  if $T_c \neq$ 0
   for each $C_i$
     $R' \leftarrow$ Replace ($R_k$)
     $T_c' \leftarrow$ Con ($R'$)
      if $T_c' < T_c$
       $T_c \leftarrow T_c'$
       $R \leftarrow R'$
      end if
   end for
  end if
return $T_c$, $R$
```

Algorithm 2. Select procedure for AGV routing.

(SP). The transport time, delay time, and conflict points are calculated, and a tabu list is generated (lines 8–17). If a solution exists in the elite solution set that conflict-free paths can be generated in path planning is marked as y_1(lines 10–13). The optimal solution in the elite solution set whose delay time is less than ε is marked as y_2 (lines 18–20). If the objective function values showed no improvement after multiple iterations are marked as y_3 (lines 24–26), the iteration of the elite solution set is updated by HDSTA by incorporating the Tabu List constraints until one of the termination conditions are met and generating an integrated scheduling solution containing the sequential assignment solution and the detail path solution.

4.1. HDSTA

The state transition algorithm (STA) (Yang et al., 2013) is a kind of intelligent optimization algorithm originally proposed

```
Require: graph $G$, set of depots
$X = \{x_1,\ x_2,\ x_3 \ldots x_s\}$
Initialize matrix $H_s$
repeat
CLOSE list $\leftarrow$ 0; final $\leftarrow$ 0; Mark $= \varnothing$
OPEN list $\leftarrow$ start
  while
    while OPEN list $\neq$ 0
      if the number of the latest node in
      OPEN list = 1
        Current node $\leftarrow$ the latest node in
        OPEN list
      else Current node $\leftarrow$ the first node
        Mark $\leftarrow$ OPEN list; Mark $\leftarrow$ CLOSE list
      end if
      if current node = end point
        break final $\leftarrow$ CLOSE list
      end if
        for each neighbor (Current node)
          if neighbor $\notin$ Obstacle
            new cost $= f$(neighbor)
              if new cost $<$ cost allel neighbor
              $\notin$ CLOSE list
                OPEN list $\leftarrow$ neighbor
              else CLOSE list $\leftarrow$ neighbor
            end if
          CLOSE list $\leftarrow$ current node
        end if
      end for
  end
  if Mark $\neq$ $\varnothing$
      OPEN list $\leftarrow$ Mark ($i$); CLOSE list $\leftarrow$
      Mark ($j$)
  else break
  end if
 end
final = sort (final)
$N_S$ $\leftarrow$ final
until the specified termination
criterion is met
return $N_S$
```

Algorithm 3. Improved *A** algorithm for establishing the path expert database.

by Zhou et al. (2012) with good global search capability and convergence property. In our proposed HDSTA, the integrated problem is decomposed into individual elements and the individual S is defined by three "state spaces" related to its tasks sequence, AGV dispatch, and the corresponding routes, as depicted in Figure 2.

The task sequence state space Q registers for each task. The space Q consists of some types of tasks, with the subspace $q_d \in Q$ representing one type of task collection. The first type is charging requirement, while the others depend on the number of heterogeneous AGV types. It is worth mentioning that, to ensure performing urgent tasks first, the sequence of the task in each subspace must observe the rule of task priority. The dispatching state space D contains the task assignment for all AGVs, and the subspace $d_k \in D$ represents the dispatching of AGV $k \in K$. Finally, the AGV routing R corresponds to all paths, while the subspace $r_k^{u_k} \in R$ represents the path of the uth task of the kth AGV, respectively.

The integrated scheduling state space of AGVs is decoded for three subproblems, namely, task sequence, dispatching, and routing. Regularly representing an individual solution with appropriate random numbers is a very effective method to solve combinatorial optimization problems. However, to solve scheduling optimally in an integrated manner, intrinsic connections and constraints between the subproblems must be established.

An illustration of the integrated method is given in Figure 3. A sequence space consists of three types of requirements: a charging request (denoted in red), a piggyback transportation requirement (denoted in yellow), and a pallet transportation requirement (denoted in green). By randomly assigning tasks in each subspace to the matching AGVs, the corresponding dispatching space is generated. Each time the generated dispatching space performs the routing procedure, that is, to select the optimal path with the least collision in the path expert database and generate the conflict result feedback C. The optimization objective result S is the sum of total travel time *Cost* based on the current solution of dispatching space and conflict result C. The feasible solution of scheduling consists of a dispatching scheme and a detailed routing scheme. The role of sequence space is to define various requirements with priority and increase the search range of the algorithm.

In the task sequence state space Q, a candidate solution set is generated by the three special operators, a swap operator, shift operator, and symmetry operator (Yang et al., 2013), which are very effective to solve discrete optimization problems. Moreover, a candidate solution set is created by the times of the transformation called the search enforcement (SE) and the translation operator is performed only if a better new trail is found.

In the AGV dispatching space, the same four operators are applied to produce a candidate solution set, which is referred to as self-learning. However, the search space of the basic state transformation algorithm is normalized or specialized and cannot directly solve the problem with multiple subspaces. In the dispatching space, the communication strategy between the subspace d_k is necessary to exchange information for increasing the search intensity. Thus, we employed two move operators, illustrated by Figure 4. (1)—the single insertion operator (SI): displaces the last task element of one subspace of dispatching to a random position in another subspace of dispatching and (2) the position-based crossover operator (PBC): exchange task elements of the same position randomly in two different subspaces of dispatching.

4.2. Select procedure

The method of AGV routing based on the path expert database is proposed for the first time. In the industrial context, collisions can be avoided by hardware, and conflict-free path planning is not our purpose as it takes a lot of computing

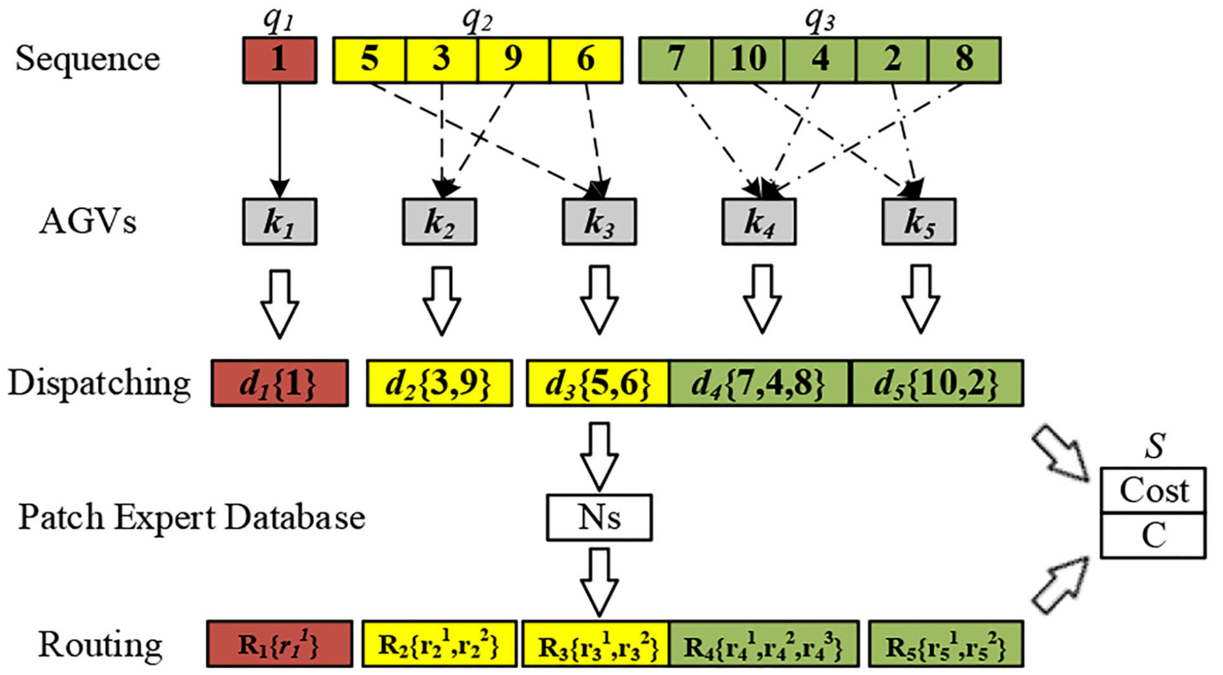


FIGURE 3
A description of the integrated scheduling problem.

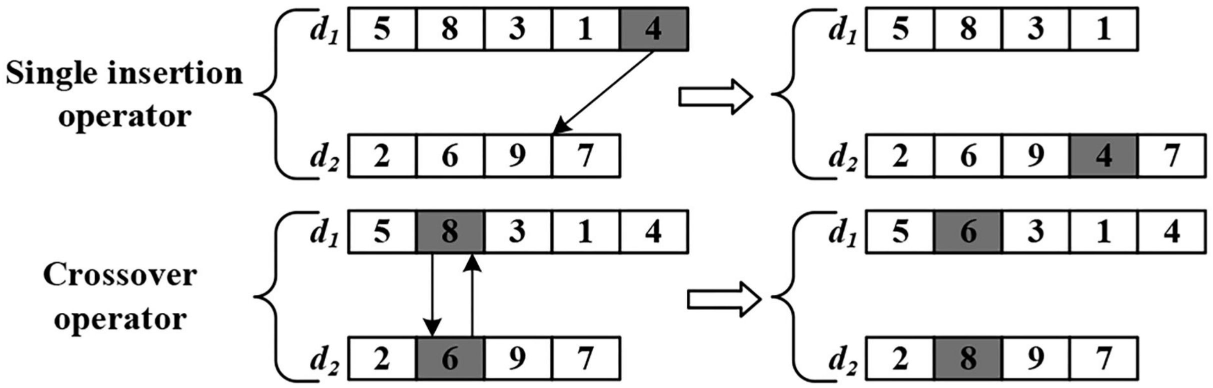


FIGURE 4
A description of move operators.

time and easily falls into locking in large-scale problems. The proposed method is to select a detailed path with the least conflicts from the path expert database according to the current dispatching and recording of the information of the conflict.

In this section, we give a path expert database N_s and task assignment for all AGVs, $D = \ d_1, d_2, d_3 \ldots d_k$ in this section. Let T_c denote the punishment time of the conflict process. As mentioned earlier, each task t consists of a pickup request o_u^t and a delivery request o_d^t. To represent a path-planning solution, we assign to the route r_t of each task two paths from the path expert database. The r_u^k and $r_{u'}^k$ are said to be traveling paths for pickup and delivery, respectively, by AGV k, and the nth path of the route r_{ij} from depot i to depot j in the path expert database is defined as $Path\left(i,\ j,\ n\right) \in \ N_s$. The selection procedure for AGV routing to optimize path planning and obtain punishment time is shown in Algorithm 2.

Set T_c to zero at the beginning (line 1). Then, a loop is executed to assign paths to each subspace d_k (lines 3–9). Each task $t \ \in \ T_k$ is assigned to the first path in the path expert database by the loop (lines 4–7). The time coordinates and punishment time of conflicts are calculated based on the time window, referred to as "CON" (line 10). When the conflict time is not zero, replace the best path with another path in the path expert database, referred to as "Replace," and update the routing data if the new path is better than all the queried paths (lines 11–20). Finally, the procedure returns the best solution (line 21).

In the dispatching space, the computational procedure does not consider the conflict situation. Thus, we establish a tabu list to record the conflict situation and then feed it back to dispatching and eliminate the unfeasible solutions in the next state to reduce conflicts whenever possible.

For example, in the current dispatching space, subspace $d_1\{2,\ 1,\ 3,\ 5\}$ and subspace $d_2\{8,\ 7,\ 6,\ 4\}$ have a conflict in the routing procedure and the conflict situation is for AGV 1 in performing task 2 and AGV 2 in performing task 8. The conflict results $d_1\{2,\ x,\ x,\ x\ \}$ and $d_2\{8,\ x,\ x,\ x\ \}$ are recorded for infeasible solution domains, where x is an arbitrary task. The result represents the infeasible solution that AGV 1 first performs task 2 while AGV 2 first performs task 8. In the following stage, the infeasible solutions are removed.

4.3. Path expert database

To the best of our knowledge, the establishment of the path expert database in the offline state for path planning is proposed for the first time. The optimal path between depots is several; besides, there are many good paths as well.

The concept of a path expert database is a collection that contains all optimal paths and good paths with sequences between depots for path replacement in case of a conflict.

The path expert database can be established by manually experience or algorithm programs depending on the different specifications of the warehouse. In this section, we propose an improved A* algorithm to generate a path expert database as shown in Algorithm 3.

The initial matrix of depots is defined by H_s (line 1). The algorithm loops over each route $r \ \in \ H_s$. A loop program calculates the paths between each depot (lines 2–33). Let the CLOSE list and the OPEN list denote a collection of nodes that have already been estimated and the collection of nodes that waiting for estimating. The path result is recorded in "final" and the points of the same valuation are recorded in "mark" (lines 3–4). The algorithm executes a loop that finds the optimal path between the two depots based on the A* algorithm and records the other points of the same valuation in each iteration (lines 6–25). If the collection "mark" is not blank, remount the information of points recorded successively to find all good paths between two depots (lines 26–29)—a record of all the path results (lines 31–32) and the procedure returns path expert database N_S finally (line 34).

5. Computational experiments

To evaluate the performance of the proposed method, computational experiments are performed in a dynamic scenario and under different scenarios with varying fleet sizes and numbers of tasks. We implemented the proposed dynamic scheduling method on a computer with an Intel (R) Core (Tm) CPU i7-9700 4.8 GHz and 8 GB RAM with a 64-bit Windows 10 operation system, while the scheduling rule is implemented in Python v3.6. The study adopts the warehouse production data located in Changsha, China. The layout of this warehouse is illustrated in Figure 5, which consists of 12 buffer area depots, 12 shop depots, 15 automatic vertical warehouse depots, and 5 charging stations. From the feedback from the practitioners, the average number of requests waiting for assigning is about 30 in a horizon, a horizon with more than 60 requests is regarded as a busy period.

The position of the depots (or stations) in the layout is fixed. Therefore, we made use of a distance matrix to compute the travel time of AGVs. We generated a path expert database through the program while offline and also note again that the collision-free trajectories are not considered in our experiments, since those collisions between the AGVs can be avoided by hardware.

5.1. Dynamic scheduling

A FlexSim-based digital simulation system is established to dynamically analyze the operation of AGV systems

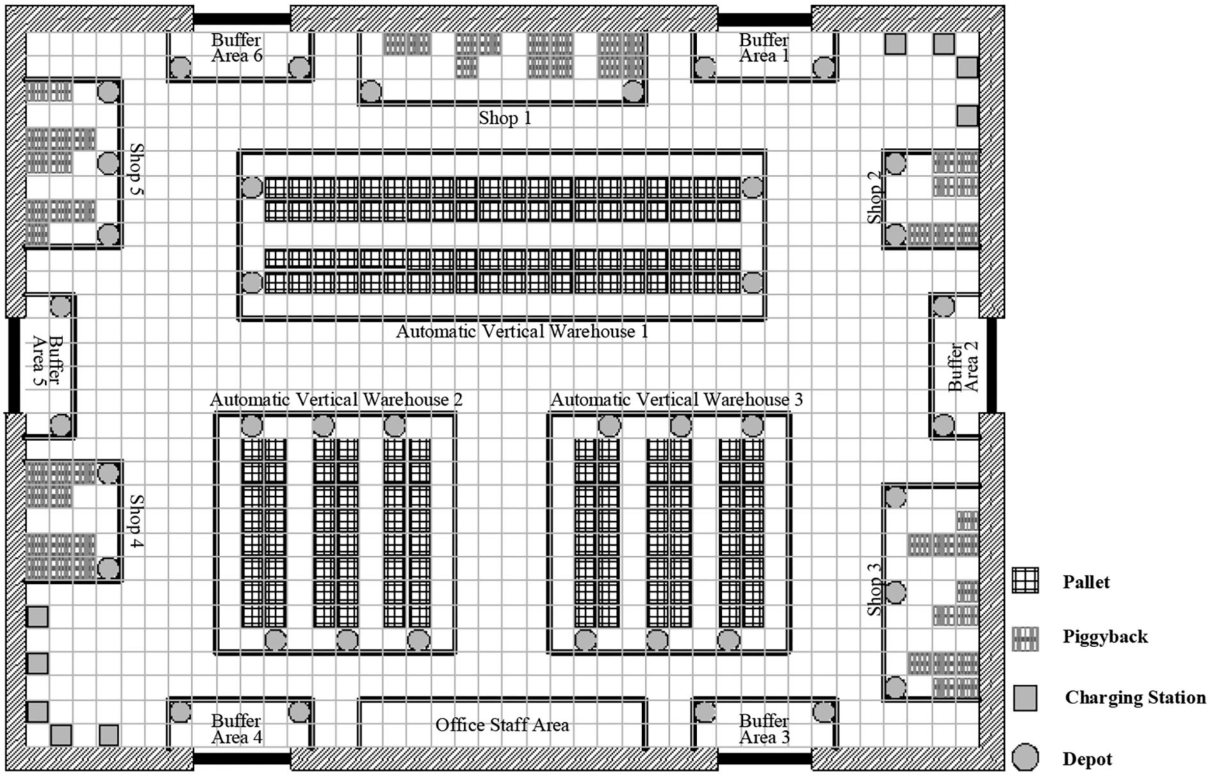


FIGURE 5
Warehouse layout.

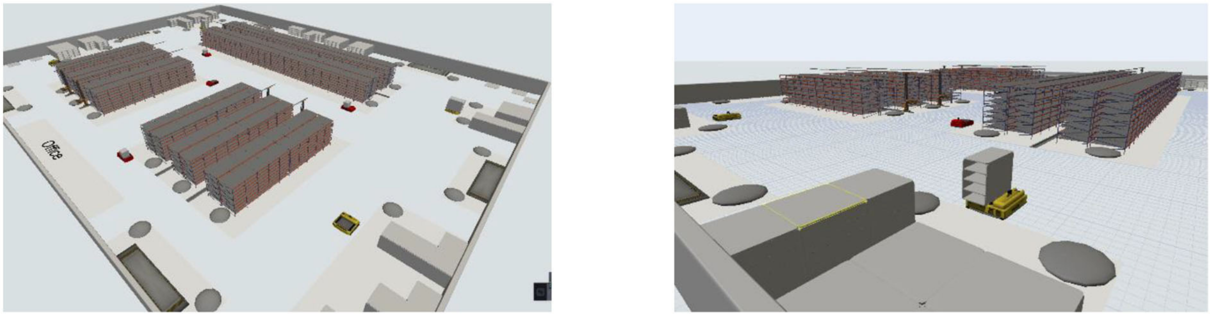


FIGURE 6
FlexSim-based digital simulation system.

under the industrial warehouse instances, as shown in Figure 6.

The description of the dynamic scheduling problem is shown in Table 1. The integrated scheduler algorithm processes a total of 60 tasks arriving at three different random intervals of time. Initially, 25 tasks are scheduled. While executing the initial schedule, 15 new tasks are added to the system at time $t = 14$ min. This results in a dynamic rescheduling of the system. While executing the current schedule, 20 more new tasks were added to the system at time $t =$ 22 min.

As stated in the methodology, this is based on the concept of scheduling and rescheduling under an appropriate time intervals methodology of dynamic scheduling. Figure 7 shows the Gantt chart of the initial schedule. The dashed line at time $t = 14$ min represents the interruption and rescheduling, the points when new tasks are added to the system. The uncompleted tasks currently at the execution stage at the interruption and the rescheduling points are task 1, task 8, task 7, and task 4. The tasks in execution will continue with the preemption in the next planning time horizon until the operation is completed. Figure 8 shows the Gantt chart for the generated new schedule, in which the new tasks are added after the interruption of the previous schedule. The operations at tasks 1, 8, 7, and 4 marked by the parallel slanted lines are the remaining operation from the previous schedule. On this schedule, all tasks in the system are either completed or the last task is under execution before the interrupt point in time $t = 22$ min. Figure 9 shows the Gantt chart for the generated new schedule. The tasks completed at the current interrupt and the rescheduling point are tasks 28, 26, 31, 33, 29, 36, 27, 37, and 34. Dynamic path planning adjusts the path without interrupting the current task execution process.

TABLE 1 The description of the dynamic scheduling problem.

Schedule	Start time (min)	New tasks
Initial	0	Task 1, Task 2, . . . , Task 25
Interrupt 1	14	Task 26, Task 27, . . . , Task 40
Interrupt 2	22	Task 41, Task 42, . . . , Task 60

5.2. Analysis of the scheduling results

The efficacy of our method is verified by computational experiments using real-world data with varying fleet sizes and numbers of tasks. The number of AGVs to be dispatched is 5,

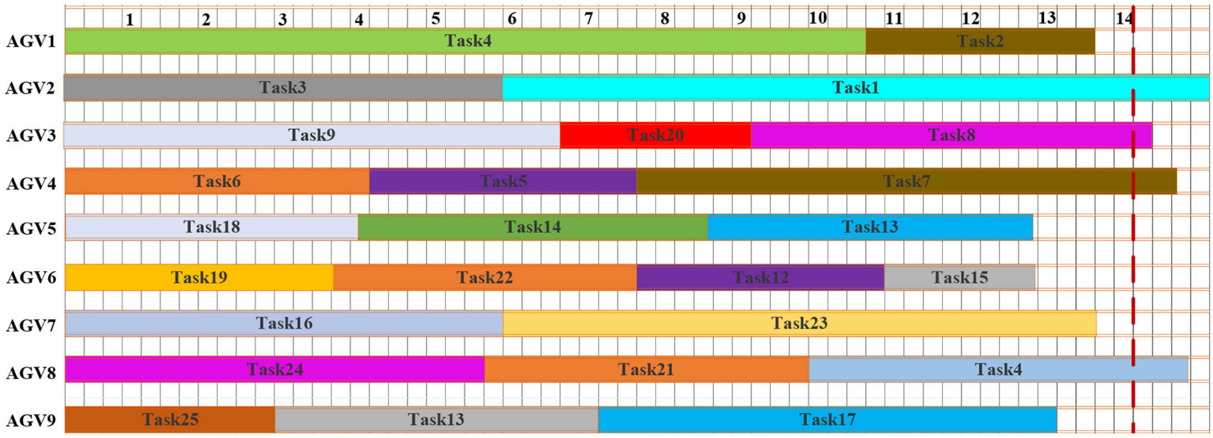


FIGURE 7
Gantt chart for dynamic scheduling 1.

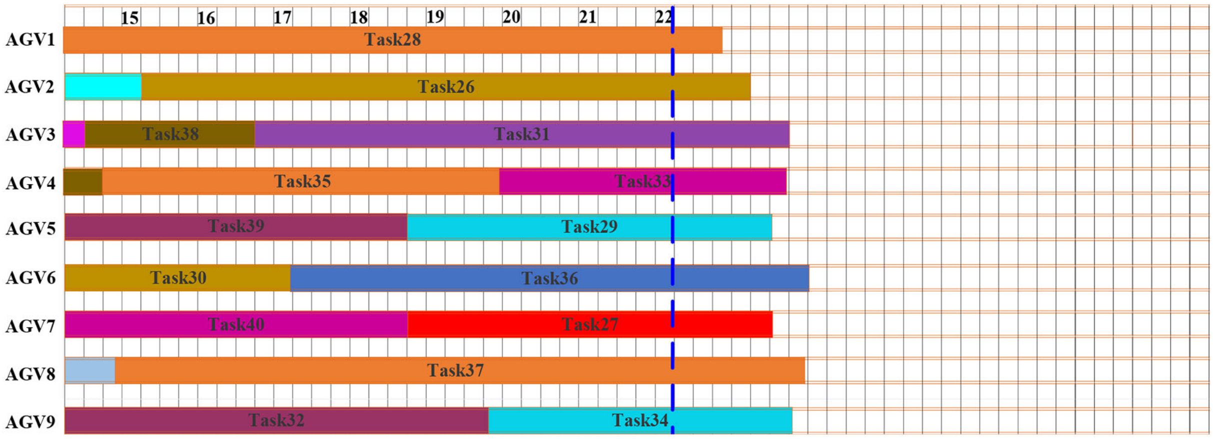


FIGURE 8
Gantt chart for dynamic scheduling 2.

10, and 15, respectively. The number of tasks to be allocated in the case is 50, 60, 70, 80, 90, and 100, respectively. Each case randomly generates five groups of tasks and runs them 10 times, for a total of 50 runs of the program. The average value is taken as the result. At present, the advanced AGV systems in industrial warehouses adopt the scheduling method of sequential optimization, of which the method proposed by Lian et al. (2020) is the most representative. Therefore, this method is selected for comparative verification of the analyses of real warehouse cases. Problems not considered in this method, such as the heterogeneity of the AGVs and battery constraint, are improved before the comparative verification in this study. In the case study, the comparison results of the task completion time and the delay time of the two methods are shown in Figures 10–12.

The results show that the integrated scheduling method proposed in this study has better performance and better solutions are found in all cases. In particular, the average task completion time is 13.62% less and the average delay time is 76.69% less than the sequential optimization of the scheduling method. The average delay time difference between the two methods is only 219 s when the number of tasks is 50 using 5 AGVs, but it increases to 3,591 s when the number of tasks increases to 100 using 15 AGVs. With the increase in task scale, the probability of conflicts between AGVs also increases dramatically. The sequential optimization scheduling method cannot avoid the impact of conflicts from the task allocation process, while the proposed integrated scheduling can avoid most conflicts by changing the task assignment and specific execution path.

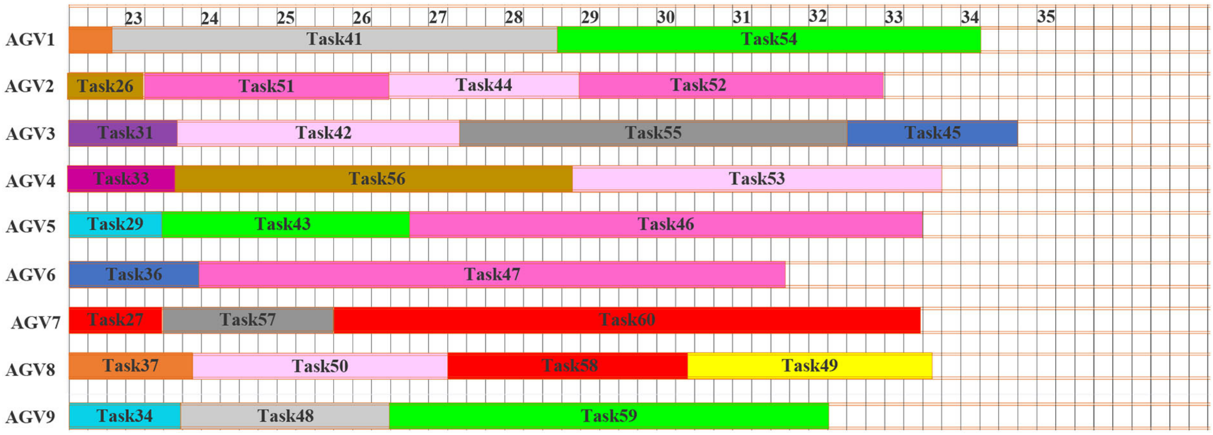


FIGURE 9
Gantt chart for dynamic scheduling 3.

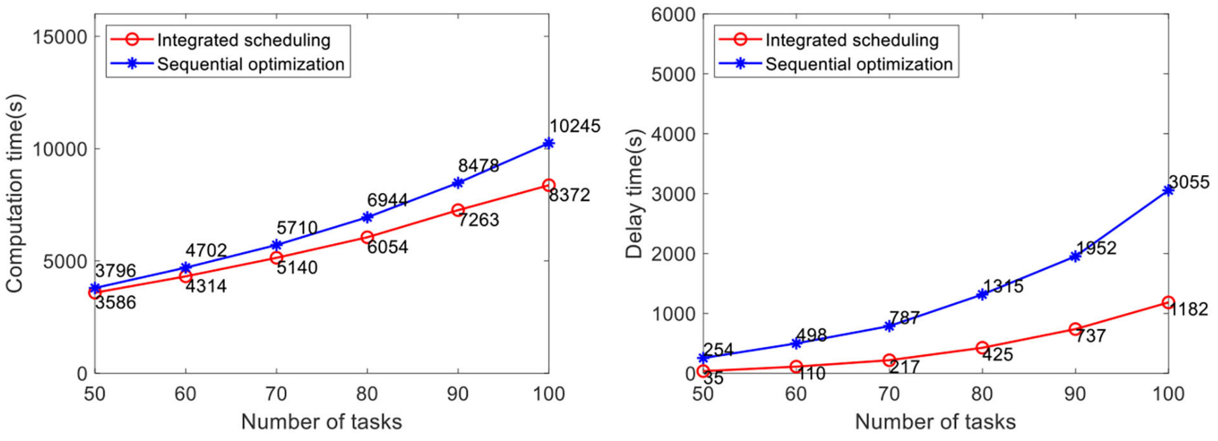


FIGURE 10
Comparative analysis of task completion time and delay time with 5 AGVs.

5.3. Comparison of algorithms

To verify the performance of HDSTA, a larger-scale case template needs to be established. We generate 10 instances for each scenario with 50–150 tasks and AGVs to be scheduled, ranging from 5 to 25; each instance runs ten times to compute the mean. In each instance, HDSTA with the adaptive large neighborhood search algorithm (HALNS) (Dang et al., 2021) was compared with the preplanning algorithm (PPA) (Maza and Castagna, 2005). HALNS is a hybrid algorithm of the adaptive large neighborhood search algorithm and the linear programming algorithm, which is proposed to solve the heterogeneous AGV scheduling problem with charge capacity constraints. However, this method does not consider the problem of conflict and deadlock. For comparison, we developed the conflict detection method to compute the delay time of its optimal solution. PPA is a strategy to generate conflict-free paths.

Table 2 compares the task completion time and scheduling computation time for varying scenarios with different numbers of AGVs and different numbers of tasks, where the task completion time is directly proportional to the operation cost, which can visually reflect the collaborative operation efficiency of AGVs, and the computation time is an important index of dynamic scheduling, which can reflect the computation efficiency of AGV systems. Table 2 compares the performance and computation time of PPA, HALNS, and HDSTA for 150 sets of tasks in 15 case types. For example, when the task volume is 50 and the number of AGVs is 5, 9, and 11, respectively, the task completion times of the scenarios calculated by optimal scheduling with the HDSTA algorithm are 3,738, 3,811, and 3,845 s and the computation times are 2.23, 2.33, and 2.37 s, respectively. The results of 150 sets

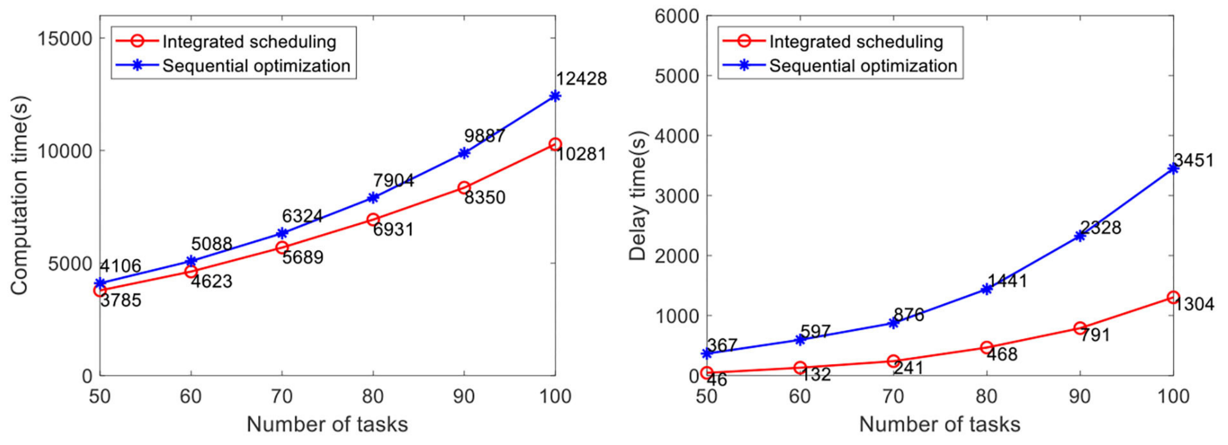


FIGURE 11
Comparative analysis of task completion time and delay time with 10 AGVs.

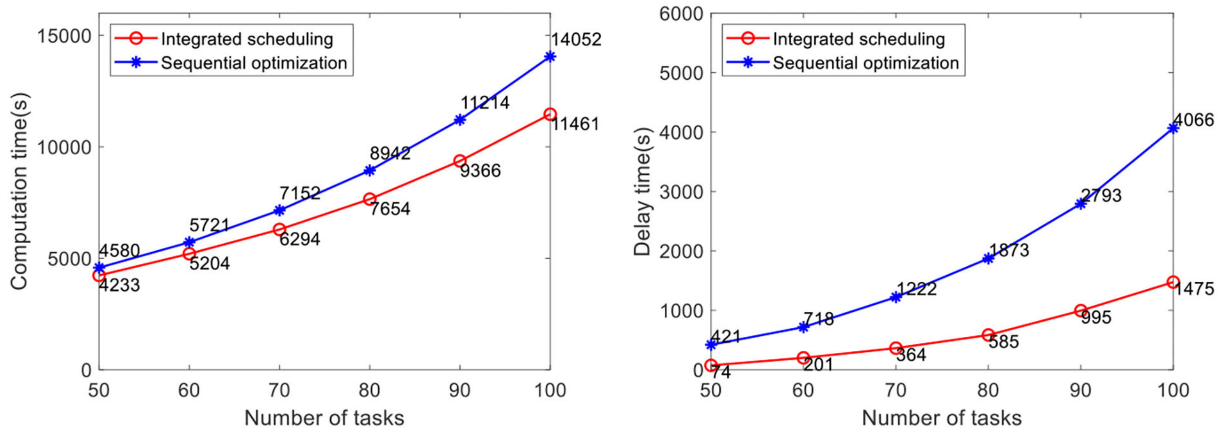


FIGURE 12
Comparative analysis of task completion time and delay time with 15 AGVs.

TABLE 2 Results of the varying scenarios.

NO.	Number of tasks	Number of AGVs	PPA		HALNS		HDSTA	
			Computation time (s)	Task completion time (s)	Computation time (s)	Task completion time (s)	Computation time (s)	Task completion time (s)
1	50	5	7.61	3,747	4.24	3,744	2.23	37,38
2	50	9	7.73	3,830	4.36	3,862	2.33	3,811
3	50	11	8.13	3,853	4.68	3,876	2.37	3,845
4	60	3	11.41	4,522	6.88	4,529	2.84	4,522
5	60	8	12.14	4,615	7.20	4,628	2.83	4,608
6	60	13	12.81	4,693	8.36	4,854	2.90	4,685
7	80	4	19.91	6,570	16.07	6,569	3.55	6,558
8	80	6	22.89	6,683	18.15	6,711	3.91	6,654
9	100	7	38.92	8,421	31.94	8,408	5.64	8,311
10	100	15	43.85	8,636	30.11	8,756	6.22	8,541
11	120	15	50.21	12,850	41.65	13,365	8.60	12,305
12	120	18	54.33	12,902	56.84	14,025	8.84	12,654
13	120	19	55.12	13,357	58.52	14,359	8.92	13,147
14	150	22	98.21	18,724	84.74	19,015	13.45	17,521
15	150	23	116.89	19,031	91.61	19,584	14.16	16,984

of tasks for 15 case types are analyzed and compared with PPA and HALNS. The average task completion time of the HDSTA solution proposed in this study is lower by 3.44 and 7.27% and the computation time is less by 84.15 and 81.92%.

6. Conclusion

This article studied the problem of scheduling a heterogeneous fleet of AGVs. A MILP model was formulated to minimize the sum of the costs of requests and the tardiness costs of conflicts. The hierarchical planning method is used to decompose the complex and integrated scheduling problem. We propose that HDSTA combine select procedures. The major novelty of this study is the ability to solve the dynamic integrated scheduling problem for heterogeneous AGV fleets with battery constraints. We performed numerical experiments to validate our model according to the real-world conditions of the automated warehouses in Changsha, China.

In the future, we may extend our research to improve our approach to multiple pickups and deliveries along the same route (multi-load AGVs) and the inclusion of path planning in the scheduling process.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

EH and SS: conceptualization and writing—original draft preparation. EH: data curation. EH and JH: methodology, validation, and formal analysis. JH: writing—review and editing and funding acquisition. All authors contributed to the article and approved the submitted version.

Funding

This study was supported by National Natural Science Foundation of China under Grant No. 61873282.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abderrahim, M., Bekrar, A., Trentesaux, D., Aissani, N., and Bouamrane, K. (2020). Bi-local search based variable neighborhood search for job-shop scheduling problem with transport constraints. *Optimiz. Lett.* 16, 255–280. doi: 10.1007/s11590-020-01674-0

Dang, Q. V., Singh, N., Adan, I., Martagan, T., and van de Sande, D. (2021). Scheduling heterogeneous multi-load AGVs with battery constraints. *Comput. Operat. Res.* 136, 105517. doi: 10.1016/j.cor.2021.105517

Desaulniers, G., Langevin, A., Riopel, D., and Villeneuve, B. (2003). Dispatching and conflict-free routing of automated guided vehicles: an exact approach. *Int. J. Flexible Manufact. Syst.* 15, 309–331. doi: 10.1023/B:FLEX.0000036032.41757.3d

Fazlollahtabar, H., and Hassanli, S. (2018). Hybrid cost and time path planning for multiple autonomous guided vehicles. *Appl. Intellig.* 48, 482–498. doi: 10.1007/s10489-017-0997-x

Fazlollahtabar, H., and Saidi-Mehrabad, M. (2015). Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. *J. Intell. Robotic Syst.* 77, 525–545. doi: 10.1007/s10846-013-0003-8

Gen, M., Zhang, W., Lin, L., and Yun, Y. (2017). Recent advances in hybrid evolutionary algorithms for multiobjective manufacturing scheduling. *Comput. Indus. Eng.* 112, 616–633. doi: 10.1016/j.cie.2016.12.045

Guo, K., Zhu, J., and Shen, L. (2020). An improved acceleration method based on multi-agent system for AGVs conflict-free path planning in automated terminals. *IEEE Access* 9, 3326–3338. doi: 10.1109/ACCESS.2020.3047916

Hamzeei, M., Farahani, R. Z., and Rashidi-Bejgan, H. (2013). An exact and a simulated annealing algorithm for simultaneously determining flow path and the location of P/D stations in bidirectional path. *J. Manufact. Syst.* 32, 648–654. doi: 10.1016/j.jmsy.2013.07.002

Hooker, J. N., and Ottosson, G. (2003). Logic-based Benders decomposition. *Mathe. Programm.* 96, 33–60. doi: 10.1007/s10107-003-0375-9

Li, G., Li, X., Gao, L., and Zeng, B. (2019). Tasks assigning and sequencing of multiple AGVs based on an improved harmony search algorithm. *J. Ambient Intell. Humaniz. Comput.* 10, 4533–4546. doi: 10.1007/s12652-018-1137-0

Li, Z., Barenji, A. V., Jiang, J., Zhong, R. Y., and Xu, G. (2020). A mechanism for scheduling multi robot intelligent warehouse system face with dynamic demand. *J. Intell. Manuf.* 31, 469–480. doi: 10.1007/s10845-018-1459-y

Lian, Y., Yang, Q., Xie, W., and Zhang, L (2020). Cyber-physical system-based heuristic planning and scheduling method for multiple automatic guided vehicles in logistics systems. *IEEE Trans. Indus. Inform.* 17, 7882–7893. doi: 10.1109/TII.2020.3034280

Lu, H., and Wang, S. (2019). A study on multi-ASC scheduling method of automated container terminals based on graph theory. *Comput. Indus. Eng.* 129, 404–416. doi: 10.1016/j.cie.2019.01.050

Lyu, X., Song, Y., He, C., Lei, Q., and Guo, W. (2019). Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems. *IEEE Access* 7, 74909–74924. doi: 10.1109/ACCESS.2019.2919109

Ma, X., Bian, Y., and Gao, F. (2020). An improved shuffled frog leaping algorithm for multiload AGV dispatching in automated container terminals. *Math. Probl. Eng.* 2020, 1260196. doi: 10.1155/2020/1260196

Maza, S., and Castagna, P. (2005). A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles. *Comput. Industry* 56, 719–733. doi: 10.1016/j.compind.2005.03.003

Nishi, T., Hiranaka, Y., and Grossmann, I. E. (2011). A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Comput. Operat. Res.* 38, 876–888. doi: 10.1016/j.cor.2010.08.012

Qiuyun, T., Hongyan, S., Hengwei, G., and Ping, W. (2021). Improved particle swarm optimization algorithm for AGV path planning. *IEEE Access* 9, 33522–33531. doi: 10.1109/ACCESS.2021.3061288

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., and Mahmoodian, V. (2015). An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Comput. Indus. Eng.* 86, 2–13. doi: 10.1016/j.cie.2015.01.003

Schiffer, M., and Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *Eur. J. Oper. Res.* 260, 995–1013. doi: 10.1016/j.ejor.2017.01.011

Singh, N., Dang, Q. V., Akcay, A., Adan, I., and Martagan, T. (2022). A matheuristic for AGV scheduling with battery constraints. *Eur. J. Oper. Res.* 298, 855–873. doi: 10.1016/j.ejor.2021.08.008

Umar, U. A., Ariffin, M. K. A., Ismail, N., and Tang, S. H. (2015). Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. *Int. J. Adv. Manuf. Technol.* 81, 2123–2141. doi: 10.1007/s00170-015-7329-2

Yang, C. H., Tang, X. L., Zhou, X. J., and Gui, W. H. (2013). A discrete state transition algorithm for traveling salesman problem. *Control Theory Applic.* 30, 1040–1046. doi: 10.7641/CTA.2013.12167

Zhang, L., Hu, Y., and Guan, Y. (2019). Research on hybrid-load AGV dispatching problem for mixed-model automobile assembly line. *Proc. CIRP* 81, 1059–1064. doi: 10.1016/j.procir.2019.03.251

Zhang, X., Sang, H., Li, J., Han, Y., and Duan, P. (2022). An effective multi-AGVs dispatching method applied to matrix manufacturing workshop. *Comput. Indus. Eng.* 163, 107791. doi: 10.1016/j.cie.2021.107791

Zhang, Y., Zhu, Z., and Lv, J. (2017). CPS-based smart control model for shopfloor material handling. *IEEE Trans. Indus. Inform.* 14, 1764–1775. doi: 10.1109/TII.2017.2759319

Zhou, X., Yang, C., and Gui, W. (2012). State transition algorithm. *J. Ind. Manag. Optim.* (2012) 8, 1039–1056. doi: 10.3934/jimo.2012.8.1039

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 09 January 2023
DOI 10.3389/fnbot.2022.1081242

Realistic Actor-Critic: A framework for balance between value overestimation and underestimation

Sicen Li[1,2], Qinyun Tang[1,2], Yiming Pang[1,2], Xinmeng Ma[1] and Gang Wang[2,3]*

[1]College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin, China, [2]Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin, China, [3]College of Shipbuilding Engineering, Harbin Engineering University, Harbin, China

Check for updates

OPEN ACCESS

EDITED BY
Huiyu Zhou,
University of Leicester,
United Kingdom

REVIEWED BY
Amit Trivedi,
University of Illinois at Chicago,
United States
Changsheng Li,
Beijing Institute of Technology, China
Zhe Min,
University College London,
United Kingdom
Ning Tan,
Sun Yat-sen University, China

*CORRESPONDENCE
Gang Wang
✉ wanggang@hrbeu.edu.cn

RECEIVED 27 October 2022
ACCEPTED 13 December 2022
PUBLISHED 09 January 2023

CITATION
Li S, Tang Q, Pang Y, Ma X and Wang G (2023) Realistic Actor-Critic: A framework for balance between value overestimation and underestimation. *Front. Neurorobot.* 16:1081242. doi: 10.3389/fnbot.2022.1081242

Introduction: The value approximation bias is known to lead to suboptimal policies or catastrophic overestimation bias accumulation that prevent the agent from making the right decisions between exploration and exploitation. Algorithms have been proposed to mitigate the above contradiction. However, we still lack an understanding of how the value bias impact performance and a method for efficient exploration while keeping stable updates. This study aims to clarify the effect of the value bias and improve the reinforcement learning algorithms to enhance sample efficiency.

Methods: This study designs a simple episodic tabular MDP to research value underestimation and overestimation in actor-critic methods. This study proposes a unified framework called Realistic Actor-Critic (RAC), which employs Universal Value Function Approximators (UVFA) to simultaneously learn policies with different value confidence-bound with the same neural network, each with a different under overestimation trade-off.

Results: This study highlights that agents could over-explore low-value states due to inflexible under-overestimation trade-off in the fixed hyperparameters setting, which is a particular form of the exploration-exploitation dilemma. And RAC performs directed exploration without over-exploration using the upper bounds while still avoiding overestimation using the lower bounds. Through carefully designed experiments, this study empirically verifies that RAC achieves 10x sample efficiency and 25% performance improvement compared to Soft Actor-Critic in the most challenging Humanoid environment. All the source codes are available at https://github.com/ihuhuhu/RAC.

Discussion: This research not only provides valuable insights for research on the exploration-exploitation trade-off by studying the frequency of policies access to low-value states under different value confidence-bounds guidance, but also proposes a new unified framework that can be combined with current actor-critic methods to improve sample efficiency in the continuous control domain.

KEYWORDS

reinforcement learning (RL), robot control, estimation bias, exploration-exploitation dilemma, uncertainty

1. Introduction

Reinforcement learning is a major tool to realize intelligent agents that can be autonomously adaptive to the environment (Namiki and Yokosawa, 2021; Yu, 2018; Fukuda, 2020). However, current reinforcement learning techniques still suffer from requiring a huge amount of interaction data, which could result in unbearable costs in real-world applications (Karimpanal and Bouffanais, 2018; Levine et al., 2018; Sutton and Barto, 2018; Dulac-Arnold et al., 2020). This study aims to mitigate this problem by better balancing exploration and exploitation.

Undesirable overestimation bias and accumulation of function approximation errors in temporal difference methods may lead to sub-optimal policy updates and divergent behaviors (Thrun and Schwartz, 1993; Pendrith and Ryan, 1997; Fujimoto et al., 2018; Chen et al., 2022). Most model-free off-policy RL methods learn approximate lower confidence bound of Q-function (Fujimoto et al., 2018; Kuznetsov et al., 2020; Lan et al., 2020; Chen et al., 2021; Lee et al., 2021) to avoid overestimation by introducing underestimation bias. However, if the lower bound has a spurious maximum, it will discourage policy to explore potentially higher uncertain regions, resulting in stochastic local-maximum and causing pessimistic underexploration (Ciosek et al., 2019). Moreover, directionally uninformed (Ciosek et al., 2019) policies, such as Gaussian policies, cannot avoid fully explored wasteful actions.

Optimistic exploration methods (Brafman and Tennenholtz, 2002; Kim et al., 2019; Pathak et al., 2019) learn upper confidence bounds of the Q-function from an epistemic uncertainty estimate. These methods are directionally informed and encourage policy to execute overestimated actions to help agents escape local optimum. However, such upper confidence bound might cause an agent to over-explore low-value regions. In addition, it increases the risk of value overestimation since transitions with high uncertainty may have higher function approximation errors to make the value overestimated. To avoid the above problems, one must carefully adjust hyperparameters and control the bias to keep the value at a balance point between lower and higher bounds: supporting stable learning while providing good exploration behaviors. We highlight that this balance is a particular form of the exploration–exploitation dilemma (Sutton and Barto, 2018). Unfortunately, most prior works have studied the overestimation and pessimistic underexploration in isolation and have ignored the under-/overestimation trade-off aspect.

We formulate the Realistic Actor-Critic (RAC), whose main idea is to learn together values and policies with different trade-offs between underestimation and overestimation in the same network. Policies guided by lower bounds control overestimation bias to provide consistency and stable convergence. Each policy guided by different upper bounds provides a unique exploration strategy to generate overestimated actions, so that the policy family can directionally explore overestimated state-action pairs uniformly and avoid over-exploration. All transitions are stored in a shared replay buffer, and all policies benefit from them to escape spurious maximum. Such a family of policies is jointly parameterized with the Universal Value Function Approximators (UVFA) (Schaul et al., 2015). The learning process can be considered as a set of auxiliary tasks (Badia et al., 2020b; Lyle et al., 2021) that help build shared state representations and sills.

However, learning such policies with diverse behaviors in a single network is challenging since policies vary widely in behavior. We introduce punished Bellman backup, which calculates uncertainty as punishment to correct value estimations. Punished Bellman backup provides fine-granular estimation control to make value approximation shift smoothly between upper and lower bounds, allowing for more efficient training. An ensemble of critics is learned to produce well-calibrated uncertainty estimations (i.e., standard deviation) on unseen samples (Amos et al., 2018; Pathak et al., 2019; Lee et al., 2021). We show empirically that RAC controls the standard deviation and the mean of value estimate bias to close to zero for most of the training. Benefiting from well-bias control, critics are trained with a high update-to-data (UTD) ratio (Chen et al., 2021) to improve sample efficiency significantly.

Empirically, we implement RAC with SAC (Haarnoja et al., 2018) and TD3 (Fujimoto et al., 2018) in continuous control benchmarks (OpenAI Gym Brockman et al., 2016, MuJoCo Todorov et al., 2012). Results demonstrate that RAC significantly improves the performance and sample efficiency of SAC and TD3. RAC outperforms the current state-of-the-art algorithms (MBPO Janner et al., 2019, REDQ Chen et al., 2021, and TQC Kuznetsov et al., 2020), achieving state-of-the-art sample efficiency on the Humanoid benchmark. We perform ablations and isolate the effect of the main components of RAC on performance. Moreover, we perform hyperparameter ablations and demonstrate that RAC is stable in practice. The higher sample efficiency allows RAC to facilitate further applications of the RL algorithm in automatic continuous control.

This study makes the following contributions:

(i) Highlighting that agents could over-explore low-value states due to inflexible under-/overestimation trade-off in the fixed hyperparameters setting, and it is a particular form of the exploration–exploitation dilemma;

(ii) Defining a unified framework called Realistic Actor-Critic (RAC), which employs Universal Value Function Approximators (UVFA) to simultaneously learn policies with different value confidence-bond with the same neural network, each with a different under-/overestimation trade-off;

(iii) Experimental evidence that the performance and sample efficiency of the proposed method are better than state-of-the-art methods on continuous control tasks.

The study is organized as follows. Section 2 describes related works and their results. Section 3 describes the problem setting and preliminaries of RL. Section 4 explains the under-/overestimation trade-off. Section 5 introduces the punished Bellman backup and RAC algorithm. Section 6 presents experimental results that show the sample efficacy and final performance of RAC. Finally, Section 7 presents our conclusions.

2. Related works

2.1. Underestimation and overestimation of Q-function

The maximization update rule in Q-learning has been shown to suffer from overestimation bias which is cited as the reason for nonlinear function approximation fails in RL (Thrun and Schwartz, 1993).

Minimizing the value ensemble is a standard method to deal with overestimation bias. Double DQN (Van Hasselt et al., 2016) was shown to be effective in alleviating this problem for discrete action spaces. Clipped double Q-learning (CDQ) (Fujimoto et al., 2018) took the minimum value between a pair of critics to limit overestimation. Maxmin Q-learning (Lan et al., 2020) mitigated the overestimation bias by using a minimization over multiple action-value estimates. However, minimizing a Q-function set cannot filter out abnormally small values, which causes undesired pessimistic underexploration problem (Ciosek et al., 2019). Using minimization to control overestimation is coarse and wasteful as it ignores all estimates except the minimal one (Kuznetsov et al., 2020).

REDQ (Chen et al., 2021) proposed in-target minimization, which used a minimization across a random subset of Q-functions from the ensemble to alleviate the above problems. REDQ (Chen et al., 2021) showed that their method reduces the standard deviation of the Q-function bias to close to zero for most of the training. Truncated Quantile Critics (TQC) (Kuznetsov et al., 2020) truncated the right tail of the distributional value ensemble by dropping several of the topmost atoms to control overestimation. Weighted bellman backup (Lee et al., 2021) and uncertainty-weighted actor-critic (Wu et al., 2021) prevent error propagation (Kumar et al., 2020) in Q-learning by reweighing sample transitions based on uncertainty estimations from the ensembles (Lee et al., 2021) or Monte Carlo dropout (Wu et al., 2021). AdaTQC (Kuznetsov et al., 2021) proposed an auto mechanism for controlling overestimation bias. Unlike prior works, our work does not reweight sample transitions but directly adds uncertainty estimations to punish the target value.

The effect of underestimation bias on learning efficiency is environment-dependent (Lan et al., 2020). Therefore, choosing suitable parameters to balance under- and overestimating for entirely different environments may be hard. This work propose to solve this problem by learning about optimistic and pessimistic policy families.

2.2. Ensemble methods

In deep learning, ensemble methods are often used to solve the two key issues, uncertainty estimations (Wen et al., 2020; Abdar et al., 2021) and out-of-distribution robustness (Dusenberry et al., 2020; Havasi et al., 2020; Wenzel et al., 2020). In reinforcement learning, using an ensemble to enhance value function estimation was widely studied, such as averaging a Q-ensemble (Anschel et al., 2017; Peer et al., 2021), bootstrapped actor-critic architecture (Kalweit and Boedecker, 2017; Zheng et al., 2018), calculating uncertainty to reweight sample transitions (Lee et al., 2021), minimization over ensemble estimates (Lan et al., 2020; Chen et al., 2021), and updating the actor with a value ensemble (Kuznetsov et al., 2020; Chen et al., 2021). MEPG (He et al., 2021) introduced a minimalist ensemble consistent with Bellman update by utilizing a modified dropout operator.

A high-level policy can be distilled from a policy ensemble (Chen and Peng, 2019; Badia et al., 2020a) by density-based selection (Saphal et al., 2020), selection through elimination (Saphal et al., 2020), choosing the action that max all Q-functions (Jung et al., 2020; Parker-Holder et al., 2020; Lee et al., 2021), Thompson sampling (Parker-Holder et al., 2020), and sliding-window UCBs (Badia et al., 2020a). Leveraging uncertainty estimations of the ensemble (Osband et al., 2016; Kalweit and Boedecker, 2017; Zheng et al., 2018) simulated training different policies with a multi-head architecture independently to generate diverse exploratory behaviors. Ensemble methods were also used to learn joint state presentation to improve sample efficiency. There were two main methods: multi-heads (Osband et al., 2016; Kalweit and Boedecker, 2017; Zheng et al., 2018; Goyal et al., 2019) and UVFA (Schaul et al., 2015; Badia et al., 2020a,b). This study uses uncertainty estimation to reduce value overestimation bias, a simple max operator to get the best policy, and learning joint state presentation with UVFA.

2.3. Optimistic exploration

Pessimistic initialization (Rashid et al., 2020) and a learning policy that maximizes a lower confidence bound value could suffer a pessimistic underexploration problem (Ciosek et al., 2019). Optimistic exploration is a promising solution to ease the above problem by applying the principle of optimism in the face of uncertainty (Brafman and Tennenholtz, 2002). Disagreement (Pathak et al., 2019) and EMI (Kim et al., 2019) considered uncertainty as intrinsic motivation to encourage agents to explore the high-uncertainty areas of the environment. Uncertainty punishment proposed in this study can also be a particular intrinsic motivation. Different from studies of Pathak et al. (2019) and Kim et al. (2019), which usually choose the weight ≥ 0 to encourage exploration, punished Bellman backup use the weight ≤ 0 to control value bias. SUNRISE (Lee et al., 2021) proposed an optimistic exploration that chooses the action that maximizes upper confidence bound (Chen et al., 2017) of Q-functions. OAC (Ciosek et al., 2019) proposed an off-policy exploration policy that is adjusted to a linear fit of upper bounds to the critic with the maximum Kullback–Leibler (KL) divergence constraining between the exploration policies and the target policy. Most importantly, our work provides a unified framework for the under-/overestimation trade-off.

3. Problem setting and preliminaries

In this section, we describe the notations and introduce the concept of maximum entropy RL.

3.1. Notation

We consider the standard reinforcement learning notation, with states $\mathbf{s}$, actions $\mathbf{a}$, reward $r(\mathbf{s},\mathbf{a})$, and dynamics $p(\mathbf{s}' \mid \mathbf{s},\mathbf{a})$. The discounted return $R_t = \sum_{k=0}^{\infty}\gamma^k r_k$ is the total accumulated rewards from timestep t, $\gamma \in [0,1]$ is a discount factor determining the priority of short-term rewards. The objective is to find the optimal policy $\pi_\phi(\mathbf{s} \mid \mathbf{a})$ with parameters ϕ, which maximizes the expected return $J(\phi) = \mathbb{E}_{p_\pi}[R_t]$.

3.2. Maximum entropy RL

The maximum entropy objective (Ziebart, 2010) encourages the robustness to noise and exploration by maximizing a weighted objective of the reward and the policy entropy:

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}_{\mathbf{s}\sim p,\mathbf{a}\sim\pi}\left[r(\mathbf{s},\mathbf{a}) + \alpha\mathcal{H}\left(\pi\left(\cdot \mid \mathbf{s}\right)\right)\right], \quad (1)$$

where α is the temperature parameter used to determine the relative importance of entropy and reward. Soft Actor-Critic (SAC) (Haarnoja et al., 2018) seeks to optimize the maximum entropy objective by alternating between a soft policy evaluation and a soft policy improvement. A parameterized soft Q-function $Q_\theta(\mathbf{s},\mathbf{a})$, known as the critic in actor-critic methods, is trained by minimizing the soft Bellman backup:

$$\begin{aligned}\mathcal{L}_{\texttt{critic}}(\theta) &= \mathbb{E}_{\tau\sim\mathcal{B}}[(Q_\theta(\mathbf{s},\mathbf{a}) - y)^2], y\\ &= r - \gamma\mathbb{E}_{\mathbf{a}'\sim\pi_\phi}[Q_{\bar{\theta}}(\mathbf{s}',\mathbf{a}') - \alpha\log\pi_\phi(\mathbf{a}' \mid \mathbf{s}')],\end{aligned} \quad (2)$$

where $\tau = (\mathbf{s},\mathbf{a},r,\mathbf{s}')$ is a transition, $\mathcal{B}$ is a replay buffer, $\bar{\theta}$ are the delayed parameters which are updated by exponential moving average $\bar{\theta} \leftarrow \rho\theta + (1-\rho)\bar{\theta}$, ρ is the target smoothing coefficient, and y is the target value. The target value $Q_{\bar{\theta}}(\mathbf{s}',\mathbf{a}')$ is obtained by using two networks $Q^1_{\bar{\theta}}(\mathbf{s}',\mathbf{a}')$ and $Q^2_{\bar{\theta}}(\mathbf{s}',\mathbf{a}')$ with minimum operator:

$$Q_{\bar{\theta}}(\mathbf{s}',\mathbf{a}') = min(Q^1_{\bar{\theta}}(\mathbf{s}',\mathbf{a}'), Q^2_{\bar{\theta}}(\mathbf{s}',\mathbf{a}')). \quad (3)$$

The parameterized policy π_ϕ, known as the actor, is updated by minimizing the following object:

$$\mathcal{L}_{\texttt{actor}}(\phi) = \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\phi}\left[\alpha\log\left(\pi_\phi\left(\mathbf{a} \mid \mathbf{s}\right)\right) - Q_\theta\left(\mathbf{a},\mathbf{s}\right)\right]. \quad (4)$$

SAC uses an automated entropy adjusting mechanism to update α with the following objective:

$$\mathcal{L}_{\texttt{temp}}(\alpha) = \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\phi}\left[-\alpha\log\pi_\phi\left(\mathbf{a} \mid \mathbf{s}\right) - \alpha\overline{\mathcal{H}}\right], \quad (5)$$

where $\overline{\mathcal{H}}$ is the target entropy.

4. Understanding under-/overestimation trade-off

This section briefly discusses the estimation bias issue and empirically shows that a better under-/overestimation trade-off may improve learning performance.

4.1. Under-/overestimation trade-off

Under-/overestimation trade-off is a special form of the exploration–exploitation dilemma. This is illustrated in Figure 1. At first, the agent starts with a policy π_{past}, trained with lower bound $\hat{Q}_{LB}(\mathbf{s},\mathbf{a})$, becoming π_{LB}. We divide the current action space into four regions:

(i) High uncertainty, low-value. Highly stochastic regions also have low values; overestimation bias might cause an agent to over-explore a low-value area;

(ii) High uncertainty, excessive errors. This region has high uncertainty but is full of unseen transitions that can have excessive-high approximation errors, which may cause catastrophic overestimation and need fewer samples;

(iii) High uncertainty, controllable errors. This region has high uncertainty and is closer to the π_{LB}, with controllable approximation errors, and needs more samples;

(iv) Full explored. Since π_{past} is gradually updated to π_{LB}, the area is fully explored and needs less samples.

To prevent catastrophic overestimation bias accumulation, SAC (Haarnoja et al., 2018), TD3 (Fujimoto et al., 2018), and REDQ (Chen et al., 2021) introduce underestimation bias to learn lower confidence bounds of Q-functions, similar to Equation 3. However, directionally uninformed policies, such as gaussian policies, will sample actions located in region

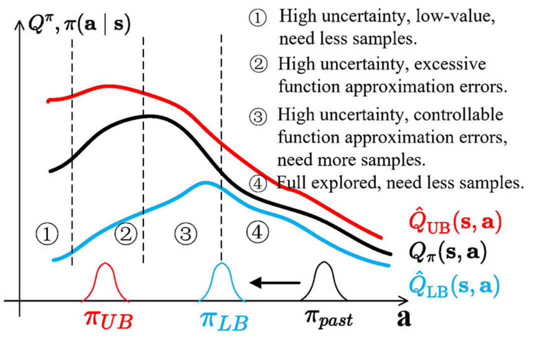

$Q^\pi, \pi(\mathbf{a} \mid \mathbf{s})$
① High uncertainty, low-value, need less samples.
② High uncertainty, excessive function approximation errors.
③ High uncertainty, controllable function approximation errors, need more samples.
④ Full explored, need less samples.
①
②
③
④
$\hat{Q}_{\text{UB}}(\mathbf{s}, \mathbf{a})$
$Q_\pi(\mathbf{s}, \mathbf{a})$
$\hat{Q}_{\text{LB}}(\mathbf{s}, \mathbf{a})$
π_{UB}
π_{LB}
π_{past}
a

FIGURE 1
Balance between value underestimation and overestimation in actor-critic methods. The state s is fixed. The graph shows Q_π **(in black)**, which is unknown to the algorithm, estimated lower bound $\hat{Q}_{LB}$ **(in blue)**, higher bound $\hat{Q}_{UB}$ **(in red)**, two policies, π_{LB} **(in blue)** and π_{past} **(in black)**, at different time steps of the algorithm, and exploration policies π_{UB} **(in red)** for optimistic exploration.

4 with half probability. If the lower bound has a spurious maximum and policies are directionally uninformed (Ciosek et al., 2019), lower bound policy π_{LB} may be stuck at the junction of regions 3 and 4. This is wasteful and inefficient, causing pessimistic underexploration.

π_{UB}, which is used in optimistic exploration methods (Brafman and Tennenholtz, 2002; Kim et al., 2019; Pathak et al., 2019), can encourage agents to execute overestimated actions and explore potential high-value regions with high uncertainty. However, regions with high and overestimated actions, such as region 2, may have excessive function approximation errors. Alternatively, if highly uncertain regions also have low values (like region 1), overestimation bias might cause an agent to over-explore a low-value region.

Ideally, the exploration policies are located in region 3 to provide better exploration behaviors and keep stable updates. There are two ways to achieve this: (1) enforcing a KL constraint between π_{UB} and π_{LB} (like OAC Ciosek et al., 2019); and (2) balancing $\hat{Q}$ between $\hat{Q}_{LB}$ and $\hat{Q}_{UB}$, and we call it an under-/overestimation trade-offs.

However, in practical applications, Q_π is unknown, and it is not easy to tune to ideal conditions through constant hyperparameters.

4.2. A simple MDP

We show this effect in a simple Markov decision process (MDP), as shown in Figure 2. Any state's optimal policy is the left action. If the agent wants to go to state 9, it must go through states 1–8 with high uncertainty and low values.

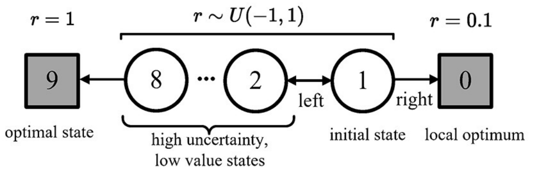


FIGURE 2
A simple episodic MDP (Lan et al., 2020), adapted from Figure 6.5 in the study of Sutton and Barto (2018). This MDP has two terminal states: state 9 and state 0. Every episode starts from state 1, which has two actions: **Left** and **Right**. The MDP is deterministic. Once the agent takes into any states, the MDP will reward back: $r = 0.1$ for terminal states 0, $r = 1$ for terminal states 9, and a reward $r \sim U(-1, 1)$ for non-terminal states 1–8. State 9 is the optimal state, state 0 is a local optimum, and states 1–8 are the high-uncertainty and low-value states.

In the experiment, we used a discount factor $\gamma = 0.9$; a replay buffer with size 5,000; a Boltzmann policy with *temperature* $= 0.1$; tabular action values with uniform noisy respect to a Uniform distribution $U(-0.1, 0.1)$, initialized with a Uniform distribution $U(-5, 5)$; and a learning rate of 0.01 for all algorithms.

The results in Figure 3 verify our hypotheses in Section 4.1. All algorithms converge, but each has a different convergence speed. $\hat{Q}_{LB}$ underestimates too much and converges to a suboptimal policy in the early learning stage, causing slow convergence. For $\beta = 0.5$ and 1.0, optimistic exploration drives the agent to escape the local optimum and learn faster. However, $\hat{Q}$ overestimates too much for $\beta = 2.0$, significantly impairing the convergence speed of the policy. In addition, no matter what parameter β takes, the agent still over-explores low-value states at different time steps (see Figure 3).

RAC avoids over-exploration in low-value states and is the fastest to converge to the optimal policy. Furthermore, RAC maintains the Q bias close to zero without catastrophic overestimation throughout the learning process, indicating that RAC keeps an outstanding balance between underestimation and overestimation.

5. Realistic Actor-Critic

We present Realistic Actor-Critic (RAC), which can be used in conjunction with the most modern off-policy actor-critic RL algorithms in principle, such as SAC (Haarnoja et al., 2018) and TD3 (Fujimoto et al., 2018). We describe only the SAC version of RAC (RAC-SAC) in the main body for the exposition. The TD3 version of RAC (RAC-TD3) follows the same principles and is fully described in Appendix B.

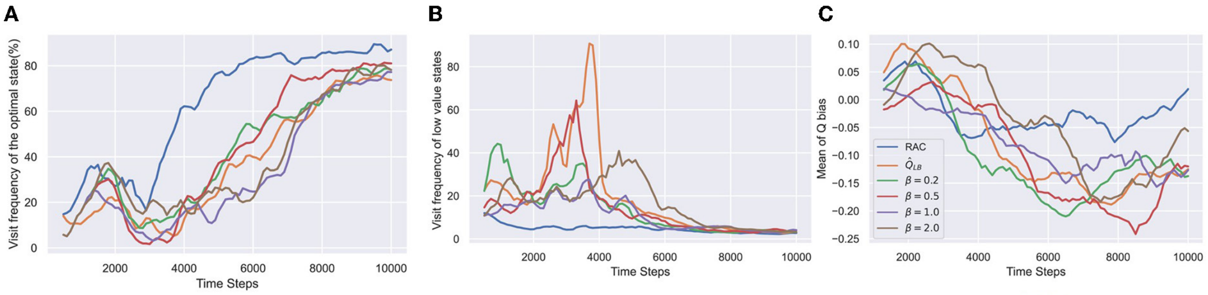


FIGURE 3
Results of the simple MDP. In-target minimization target from REDQ is used as $\hat{Q}_{LB}$. $\hat{Q}_b = mean(\hat{Q}'_{LB}) + \beta standarddeviation(\hat{Q}'_{LB})$ can perform optimistic exploration. β is a key parameter to control value bias. If $\beta = 0$, $\hat{Q}_b$ is equal to $\hat{Q}_{LB}$. As β increases, $\hat{Q}_b$ gradually approaches $\hat{Q}_{UB}$. The horizontal axis indicates the number of time steps. **(A)** Visit frequency of the optimal state is the ratio of the frequency of visiting the optimal state among all termination states. The higher the value, the lower the probability that the agent is stuck in a local optimum. **(B)** Visit frequency of low-value states is the ratio of the visit frequency of low-value state 2–8 and the optimal state 9. The lower the value, the fewer steps the agent wastes in low-value states. This value has been subtracted by 7, as the minimum step size to reach the optimum state is seven. **(C)** Q bias measures the difference between the estimated Q values and true Q values. All results are estimated by the Monte Carlo method and averaged over eight seeds.

5.1. Punished Bellman backup

Punished Bellman backup is a variant of soft Bellman backup (Equation 2). The idea is to maintain an ensemble of N soft Q-functions $Q_{\theta_i}(\mathbf{s}, \mathbf{a})$, where θ_i denotes the parameters of the $i - th$ soft Q-function, which are initialized randomly and independently for inducing an initial diversity in the models (Osband et al., 2016), but updated with the same target.

Given a transition τ_t, punished Bellman backup considers following punished target y:

$$y = r_t + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi_\phi}[\bar{Q}_{\bar{\theta}}(\mathbf{s}', \mathbf{a}') - \beta \hat{s}(Q_{\bar{\theta}}(\mathbf{s}', \mathbf{a}')) \\ - \alpha \log \pi_\phi\left(\mathbf{a}' \mid \mathbf{s}'\right)], \tag{6}$$

where $\bar{Q}_{\bar{\theta}}(\mathbf{s}, \mathbf{a})$ is the sample mean of the target Q-functions and $\hat{s}(Q_{\bar{\theta}}(\mathbf{s}, \mathbf{a}))$ is the sample standard deviation of target Q-functions with bessel's correction (Warwick and Lininger, 1975). Punished Bellman backup uses $\hat{s}(Q_{\bar{\theta}}(\mathbf{s}, \mathbf{a}))$ as uncertainty estimation to punish value estimation. $\beta \geq 0$ is the weighting of the punishment. Note that we do not propagate gradient through the uncertainty $\hat{s}(Q_{\bar{\theta}}(\mathbf{s}, \mathbf{a}))$.

We write $Q^i_{\mathbf{sa}}$ instead of $Q_{\theta_i}(\mathbf{s}, \mathbf{a})$ and $Q^i_{\mathbf{s}'\mathbf{a}'}$ instead of $Q_{\theta_i}(\mathbf{s}', \mathbf{a}')$ for compactness. Assuming each Q-function has random approximation error $e^i_{\mathbf{sa}}$ (Thrun and Schwartz, 1993; Lan et al., 2020; Chen et al., 2021), which is a random variable belonging to some distribution,

$$Q^i_{\mathbf{sa}} = Q^*_{\mathbf{sa}} + e^i_{\mathbf{sa}}, \tag{7}$$

where $Q^*_{\mathbf{sa}}$ is the ground truth of Q-functions. M is the number of actions applicable at state $\mathbf{s}'$. The estimation bias Z_{MN} for a transition τ_t is defined as

$$Z_{MN} \overset{\text{def}}{=} \left[r + \gamma \max_{\mathbf{a}'}(Q^{mean}_{\mathbf{s}'\mathbf{a}'} - \beta Q^{std}_{\mathbf{s}'\mathbf{a}'})\right] - \left(r + \gamma \max_{\mathbf{a}'} Q^*_{\mathbf{s}'\mathbf{a}'}\right) \\ = \gamma \left[\max_{\mathbf{a}'}(Q^{mean}_{\mathbf{s}'\mathbf{a}'} - \beta Q^{std}_{\mathbf{s}'\mathbf{a}'}) - \max_{\mathbf{a}'} Q^*_{\mathbf{s}'\mathbf{a}'}\right], \tag{8}$$

where

$$Q^{mean}_{\mathbf{s}'\mathbf{a}'} \approx \frac{1}{N}\sum_{i=1}^{N} Q^i_{\mathbf{s}'\mathbf{a}'} = \frac{1}{N}\sum_{i=1}^{N}(Q^*_{\mathbf{s}'\mathbf{a}'} + e^i_{\mathbf{s}'\mathbf{a}'}) = Q^*_{\mathbf{s}'\mathbf{a}'} \\ + \frac{1}{N}\sum_{i=1}^{N} e^i_{\mathbf{s}'\mathbf{a}'} = Q^*_{\mathbf{s}'\mathbf{a}'} + \bar{e}_{\mathbf{s}'\mathbf{a}'}, \tag{9}$$

$$Q^{std}_{\mathbf{s}'\mathbf{a}'} \approx \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(Q^i_{\mathbf{s}'\mathbf{a}'} - Q^{mean}_{\mathbf{s}'\mathbf{a}'}\right)^2} \\ = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(Q^*_{\mathbf{s}'\mathbf{a}'} + e^i_{\mathbf{s}'\mathbf{a}'} - Q^*_{\mathbf{s}'\mathbf{a}'} + \bar{e}_{\mathbf{s}'\mathbf{a}'}\right)^2} \\ = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(e^i_{\mathbf{s}'\mathbf{a}'} - \bar{e}_{\mathbf{s}'\mathbf{a}'}\right)^2} = \hat{s}(e_{\mathbf{s}'\mathbf{a}'}). \tag{10}$$

Then,

$$Z_{MN} \approx \gamma \left[\max_{\mathbf{a}'}(Q^*_{\mathbf{s}'\mathbf{a}'} + \bar{e}_{\mathbf{s}'\mathbf{a}'} - \beta \hat{s}(e_{\mathbf{s}'\mathbf{a}'})) - \max_{\mathbf{a}'} Q^*_{\mathbf{s}'\mathbf{a}'}\right]. \tag{11}$$

If one could choose $\beta = \frac{\bar{e}_{\mathbf{s}'\mathbf{a}'}}{\hat{s}(e_{\mathbf{s}'\mathbf{a}'})}$, $Q^i_{\mathbf{sa}}$ will be resumed to $Q^*_{\mathbf{sa}}$, then Z_{MN} can be reduced to near 0. However, it's hard to adjust a suitable constant β for various state-action pairs actually. We

```
Initialize actor network $\phi$, $N$ critic networks $\theta_i, i = 1, \ldots, N$, temperature network $\psi$, empty replay buffer $\mathcal{B}$, target network $\bar{\theta}_i \longleftarrow \theta_i$, for $i = 1, 2, \ldots, N$, uniform distribution $U_1$ and $U_2$
for each iteration do
    // OPTIMISTIC EXPLORATION
    execute an action $a \sim \pi_\phi(\cdot \mid s, \beta), \beta \sim U_2$.
    Observe reward $r_t$, new state $s'$
    Store transition tuple $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, a, r_t, s')\}$
    for $G$ updates do
        // UPDATE CRITICS via PUNISHED BELLMAN BACKUP
        Sample random minibatch:
        $\{\tau_j\}_{j=1}^{B} \sim \mathcal{B}$, $\{\beta_m\}_{m=1}^{B} \sim U_1$
        Compute the Q target (Equation 13)
        for $i = 1, \ldots, N$ do
            Update $\theta_i$ by minimize $\mathcal{L}_{\text{critic}}^{\text{RAC}}$ (Equation 13)
            Update target networks:
            $\bar{\theta}_i \leftarrow \rho\bar{\theta}_i + (1 - \rho)\theta_i$
    // UPDATE ACTORS AND TEMPERATURES ACCORDING TO $U_1$
    Update $\phi$ by minimize $\mathcal{L}_{\text{actor}}^{\text{RAC-SAC}}$ (Equation 14)
    Update $\psi$ by minimize $\mathcal{L}_{\text{temp}}^{\text{RAC}}$ (Equation 12)
```

Algorithm 1. RAC: SAC version.

develop vanilla RAC, which uses a constant β Appendix B.3, to research this problem.

For $\beta = 0$, the update is simple average Q-learning which causes overestimation bias (Chen et al., 2021). As β increases, increasing penalties $Q^{std}_{\mathbf{s}'\mathbf{a}'}$ decrease $E[Z_{MN}]$ gradually and encourage Q-functions to transit smoothly from higher bounds to lower bounds.

5.2. Realistic Actor-Critic agent

We demonstrate how to use punished Bellman backup to incorporate various bounds of value approximations into a full agent that maintains diverse policies, each with a different under-/overestimation trade-off. The pseudocode for RAC-SAC is shown in Algorithm 1.

RAC uses UVFA (Schaul et al., 2015) to extend the critic and actor as $Q_{\theta_i}(\mathbf{s}, \mathbf{a}, \beta)$ and $\pi_\phi\left(\cdot \mid \mathbf{s}', \beta\right)$, U_1 is a uniform training distribution $\mathcal{U}[0, a]$, a is a positive real number, and $\beta \sim U_1$ that generates various bounds of value approximations.

An independent temperature network α_ψ parameterized by ψ is used to accurately adjust the temperature with respect to β, which can improve the performance of RAC. Then, the objective (Equation 5) becomes:

$$\mathcal{L}_{\text{temp}}^{\text{RAC}}(\psi) = \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\mathbf{a}\sim\pi_\phi,\beta\sim U_1}[-\alpha_\psi(\beta)\log\pi_\phi(\mathbf{a} \mid \mathbf{s}, \beta) - \alpha_\psi(\beta)\overline{\mathcal{H}}]. \tag{12}$$

The extended Q-ensemble use punished Bellman backup to simultaneously approximate a soft Q-function family:

$$\begin{aligned} \mathcal{L}_{\text{critic}}^{\text{RAC}}(\theta_i) &= \mathbb{E}_{\tau\sim\mathcal{B},\beta\sim U_1}[\left(Q_{\theta_i}(\mathbf{s}, \mathbf{a}, \beta) - y\right)^2], \\ y &= r + \gamma\mathbb{E}_{\mathbf{a}'\sim\pi_\phi}[\bar{Q}_{\bar{\theta}}(\mathbf{s}', \mathbf{a}', \beta) - \beta\hat{s}(Q_{\bar{\theta}}(\mathbf{s}', \mathbf{a}', \beta)) \\ &\quad - \alpha_\psi(\beta)\log\pi_\phi(\mathbf{a}' \mid \mathbf{s}', \beta)] \end{aligned} \tag{13}$$

where $\bar{Q}_{\bar{\theta}}(\mathbf{s}, \mathbf{a}, \beta)$ is the sample mean of target Q-functions and $\hat{s}(Q_{\bar{\theta}}(\mathbf{s}, \mathbf{a}, \beta))$ is the corrected sample standard deviation of target Q-functions.

The extended policy π_ϕ is updated by minimizing the following object:

$$\begin{aligned} \mathcal{L}_{\text{actor}}^{\text{RAC-SAC}}(\phi) = \mathbb{E}_{\mathbf{s}\sim\mathcal{B},\beta\sim U_1}[\mathbb{E}_{\mathbf{a}\sim\pi_\phi}[\alpha_\psi(\beta)\log\left(\pi_\phi(\mathbf{a} \mid \mathbf{s}, \beta)\right) \\ - \bar{Q}_\theta(\mathbf{a}, \mathbf{s}, \beta)]], \end{aligned} \tag{14}$$

where $\bar{Q}_\theta(\mathbf{a}, \mathbf{s}, \beta)$ is the sample mean of Q-functions.

A larger UTD ratio G improves sample utilization. We find that a smaller replay buffer capacity slightly improves the sample efficiency of RAC in Section 6.5.

Note that we find that applying different samples, which are generated by binary masks from the Bernoulli distribution (Osband et al., 2016; Lee et al., 2021), to train each Q-function will not improve RAC performance in our experiments; therefore, RAC does not apply this method.

5.2.1. RAC circumvents direct adjustment

RAC leaners with a distribution of β instead of a constant β. One could evaluate the policy family to find the best β. We employ a discrete number H of values $\{\beta_i\}_{i=1}^{H}$ (see details in Appendix A.1) to implement a distributed evaluation for computational efficiency and apply the max operator to get best β.

5.2.2. Optimistic exploration

When interacting with the environment, we propose to sample β uniformly from a uniform explore distribution $U_2 = \mathcal{U}[0, b]$, where $b < a$ is a positive real number, to get optimistic exploratory behaviors to avoid pessimistic underexploration (Ciosek et al., 2019).

5.3. How RAC solves the under-/overestimation trade-off

Similar to the idea of NGU (Badia et al., 2020b), RAC decouples exploration and exploitation policies. RAC uses

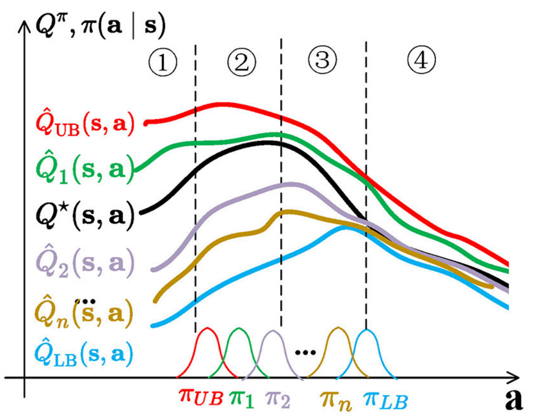

$Q^\pi, \pi(\mathbf{a} \mid \mathbf{s})$
①
②
③
④
$\hat{Q}_{\mathrm{UB}}(\mathbf{s}, \mathbf{a})$
$\hat{Q}_1(\mathbf{s}, \mathbf{a})$
$Q^\star(\mathbf{s}, \mathbf{a})$
$\hat{Q}_2(\mathbf{s}, \mathbf{a})$
$\hat{Q}_n(\mathbf{s}, \mathbf{a})$
$\hat{Q}_{\mathrm{LB}}(\mathbf{s}, \mathbf{a})$
π_{UB} π_1 π_2 π_n π_{LB}
a

FIGURE 4
Visualization of RAC. The serial numbers in the figure correspond to Section 4.1 and Figure 1. For better illustration, $\hat{Q}$ is discretized. In fact, $\hat{Q}_n$ learned by RAC is infinite and changes continuously. $Q^\star(s, a)$ is the optimal Q-function that is unknown. Q-functions are distributed between $\hat{Q}_{UB}$ and $\hat{Q}_{LB}$ and their policies are distributed between π_{UB} and π_{LB}. π_{UB}, π_1, and π_2 are used as exploration policies.

UVFA to simultaneously learn policies with the same neural network, each with different trade-offs between underestimation and overestimation. Using UVFA to learn different degrees of confidence bounds allows us to learn a powerful representation and set of skills that can be quickly transferred to the expected policy. With punished Bellman backup, RAC has a larger number of policies and values that change smoothly, allowing for more efficient training.

This is illustrated in Figure 4. Q-functions that are close to $\hat{Q}_{LB}$ (like $\hat{Q}_n$) control overestimation bias to provide consistency and stable convergence. Exploration policies (such as π_{UB}, π_1, and π_2) are far from the spurious maximum of $\hat{Q}_{LB}$ and $\hat{Q}_n$, and overestimated actions sampled from them located in regions 1, 2, and 3 lead to a quick correction to the critic estimate. All transitions are stored in a shared replay buffer, and all policies benefit from them to escape spurious maximums. Since exploration policies are not symmetric to the mean of π_{LB} and π_n, RAC also avoids directional uninformedness.

Although RAC cannot always keep the exploration policies located in region 3, the policy family avoids all behaviors concentrated in region 1 or 2. Exploration behaviors uniformly distribute in regions 1, 2, and 3, preventing over-exploration in any area.

Moreover, such policies could be quite different from a behavior standpoint and generate varied action sequences to visit unseen state-action pairs following the principle of optimism in the face of uncertainty (Chen et al., 2017; Ciosek et al., 2019; Lee et al., 2021).

6. Experiments

We designed our experiments to answer the following questions:

- Can the Realistic Actor-Critic outperform state-of-the-art algorithms in continuous control tasks?
- Can the Realistic Actor-Critic better balance between value overestimation and underestimation?
- What is the contribution of each technique in the Realistic Actor-Critic?

6.1. Setups

We implement RAC with SAC and TD3 as RAC-SAC and RAC-TD3 (see Appendix B).

The baseline algorithms are REDQ (Chen et al., 2021), MBPO (Janner et al., 2019), SAC (Haarnoja et al., 2018), TD3 (Fujimoto et al., 2018), and TQC (Kuznetsov et al., 2020). All hyperparameters we used for evaluation are the same as those in the original articles. For MBPO (https://github.com/JannerM/mbpo), REDQ (https://github.com/watchernyu/REDQ), TD3 (https://github.com/sfujim/TD3), and TQC (https://github.com/SamsungLabs/tqc_pytorch), we use the authors' code. For SAC, we implement it following the study of Haarnoja et al. (2018), and the results we obtained are similar to previously reported results. TQC20 is a variant of TQC with UTD $G = 20$ for a fair comparison.

We compare baselines on six challenging continuous control tasks (Walker2d, HalfCheetah, Hopper, Swimmer, Ant, and Humanoid) from MuJoCo environments (Todorov et al., 2012) in the OpenAI gym benchmark (Brockman et al., 2016).

The time steps for training instances on Walker2d, Hopper, and Ant are 3×10^5, and 1×10^6 for Humanoid and HalfCheetah. All algorithms explore with a stochastic policy but use a deterministic policy for evaluation similar to those in SAC. We report the mean and standard deviation across eight seeds.

For all algorithms, we use a fully connected network with two hidden layers and 256 units per layer, with Rectified Linear Unit in each layer (Glorot et al., 2011), for both actor and critic. All the parameters are updated by the Adam optimizer (Kingma and Ba, 2014) with a fixed learning rate. All algorithms adopt almost the same NN architecture and hyperparameter.

For all experiments, our learning curves show the total undiscounted return.

Using the Monte Carlo method, we estimate the mean and standard deviation of normalized Q-function bias (Chen et al., 2021) as the main analysis indicators to analyze the

TABLE 1 Performance on OpenAI gym.

	RAC-SAC	RAC-TD3	REDQ	MBPO	TQC20	TD3	SAC	TQC
Humanoid	**11,107 ± 475**	9,321 ± 1,126	5,504 ± 120	5,162 ± 350	7,053 ± 857	7,014 ± 643	7,681 ± 1,118	10,731 ± 1,296
Ant	6,283 ± 549	6,470 ± 165	5,475 ± 890	5,281 ± 699	4,722 ± 567	**6,796 ± 277**	6,433± 332	6,402± 1,371
Walker	**5,860 ± 440**	5,114 ± 489	5,034 ± 711	4,864 ± 488	5,109 ± 696	4,419 ± 1,682	5,249 ± 554	5,821 ± 457
Hopper	3,421 ± 483	3,495 ± 672	**3,563 ± 94**	3,280 ± 455	3,208 ± 538	3,433 ± 321	2,815 ± 585	3,011 ± 866
HalfCheetah	15,717 ± 1,063	15,083 ± 1,113	10,802 ± 1,179	13,477 ± 443	12,123 ± 2,600	14,462 ± 1,982	16,330 ± 323	**17,245 ± 293**
Swimmer	**143 ± 6.8**	71 ± 83	98 ± 31	-	143 ± 9.6	53 ± 8.8	51 ± 4.2	65 ± 5.8

The maximum value for each task is bolded. ± corresponds to a single standard deviation over eight runs. The best results are indicated in bold. Results of SAC, TD3, and TQC are obtained at 6×10^6 time steps for Humanoid and HalfCheetah and 3×10^6 time steps for other environments. Results of RAC, REDQ, and TQC20 are obtained at 1×10^6 time steps for Humanoid and HalfCheetah and 3×10^5 time steps for other environments. Results of MBPO are obtained at 3×10^5 time steps for Ant, Humanoid, and Walker2d, 4×10^5 for HalfCheetah and 1.25×10^5 for Hopper.

TABLE 2 Sample-efficiency comparison.

	RAC-SAC	REDQ	MBPO	TQC	TQC20	REDQ/RAC-SAC	MBPO/RAC-SAC	TQC/RAC-SAC	TQC20/RAC-SAC
Humanoid at 2,000	63 K	109 K	154 K	145 K	147 K	1.73	2.44	2.30	2.33
Humanoid at 5,000	134 K	250 K	295 K	445 K	258 K	1.87	2.20	3.32	1.93
Humanoid at 10,000	552 K	-	-	3,260 K	-	-	-	5.91	-
Ant at 1,000	21 K	28 K	62 K	185 K	42 K	1.33	2.95	8.81	2.00
Ant at 3,000	56 K	56 K	152 K	940 K	79K	1.00	2.71	16.79	1.41
Ant at 6,000	248 K	-	-	3,055 K	-	-	-	12.31	-
Walker at 1,000	27 K	42 K	54 K	110 K	50 K	1.56	2.00	4.07	1.85
Walker at 3,000	53 K	79 K	86 K	270 K	89K	1.49	1.62	10.75	1.68
Walker at 5,000	147 K	272 K	-	960 K	270 K	1.85	-	6.53	1.84

Sample efficiency (Chen et al., 2021; Dorner, 2021) is measured by the ratio of the number of samples collected when RAC and some algorithms reach the specified performance. The last four rows show how many times RAC is more sample efficient than other algorithms in achieving that performance.

value approximation quality (described in Appendix A). The average bias lets us know whether Q_θ is overestimated or underestimated, while standard deviation measures whether Q_θ is overfitting.

Sample efficiency (SE) (Chen et al., 2021; Dorner, 2021) is measured by the ratio of the number of samples collected when RAC and some algorithms reach the specified performance. Hopper is not in the comparison object as the performance of algorithms is almost indistinguishable.

6.2. Comparative evaluation

6.2.1. OpenAI gym

Figure 5 and Table 1 show learning curves and performance comparison. RAC consistently improves the performance of SAC and TD3 across all environments and performs better than other algorithms. In particular, RAC learns significantly faster for Humanoid and has better asymptotic performance for Ant, Walker2d, and HalfCheetah. RAC yields a much smaller variance than SAC and TQC, indicating that the optimistic exploration helps the agents escape from bad local optima.

6.2.2. Sample-efficiency comparison

Table 2 shows the sample-efficiency comparison with baselines. Compared with TQC, RAC-SAC reaches 3,000 and 6,000 for Ant with 16.79x and 12.31x sample efficiency, respectively. RAC-SAC performs 1.5x better than REDQ halfway through training and 1.8x better at the end of training in Walker and Humanoid. They show that a better under-/overestimation trade-off can achieve better sample-efficiency performance than the MuJoCo environments' state-of-the-art algorithms.

6.2.3. Value approximation analysis

Figure 6 presents the results for Ant, Humanoid, and Walker2d.

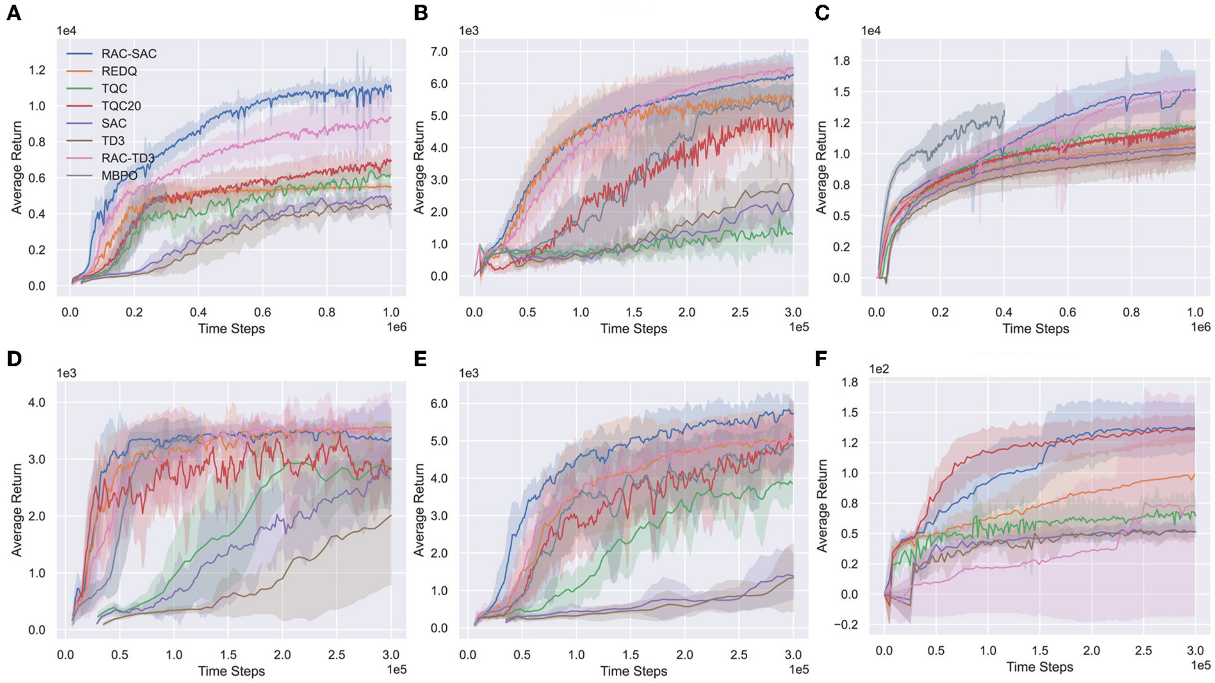


FIGURE 5
Learning curves on six Mujoco environments. The horizontal axis indicates the number of time steps. The vertical axis shows the average undiscounted return. The shaded areas denote one standard deviation over eight runs. **(A)** Humanold, **(B)** Ant, **(C)** HalfCheetah, **(D)** Hopper, **(E)** Walker2d, and **(F)** Swimmer.

In Ant and Walker2d, TQC20 has a high normalized mean of bias, indicating that TQC20 prevents catastrophic overestimation failure accumulation. TQC20 also has a high normalized standard deviation of bias, indicating that the bias is highly non-uniform, which can be detrimental. Since distributional RL is prone to overfitting with few samples, it may not be appropriate to use a high UTD ratio for TQC. In Humanoid, which has a high-dimensional state, overfitting still exists but has been alleviated.

Relative to TQC and TQC20, REDQ and RAC-SAC have a very low normalized standard deviation of bias for most of the training, indicating the bias across different state-action pairs is about the same. Thus, the Q-estimation of REDQ is too conservative in Humanoid, and the large negative bias makes REDQ trapped in a bad locally optimal policy, suffering from pessimistic underexploration. For Ant and Walker2d, although this poor exploration does not harm the performance of the policy, it still slows down convergence speed compared to RAC.

Relative to REDQ, RAC-SAC keeps the Q bias nearly zero without overestimation accumulation; this benign overestimation bias significantly improves performance. RAC-SAC strikes a good balance between overestimation bias (good performance without being trapped in a bad local optimum) and underestimation bias (slight overestimation bias and consistently small standard deviation of bias).

6.3. Why Humanoid is hard for most baselines?

Figure 7 visualizes the performance with respect to various value confidence bounds. Humanoid is extremely sensitive to the value bias. The huge state-action space of Humanoid leads to a large approximation error of the value function with small samples. The approximate lower bound inevitably has spurious maxima, while a small overestimated bias can seriously destabilize updates. It is hard to choose appropriate confidence bound for Humanoid by tuning the hyperparameters, resulting in a difficult under-/overestimation trade-off.

Algorithms (like REDQ) that rely on constant hyperparameters to control the value bias have to conservatively introduce a large underestimation error (Figure 6) to stabilize updates, leading the policy to fall into pessimistic underexploration. In contrast, other algorithms (such as TQC20) plagued by overestimation and overfitting require more samples.

Compared to Humanoid, the state-action space of other environments is much smaller. The approximate Q-functions can easily fit the true Q values accurately, significantly reducing the possibility of spurious maxima. Therefore, optimistic exploration may not be a required component for these environments. So, we can see that they are not very sensitive to various value confidence bounds from Figure 7.

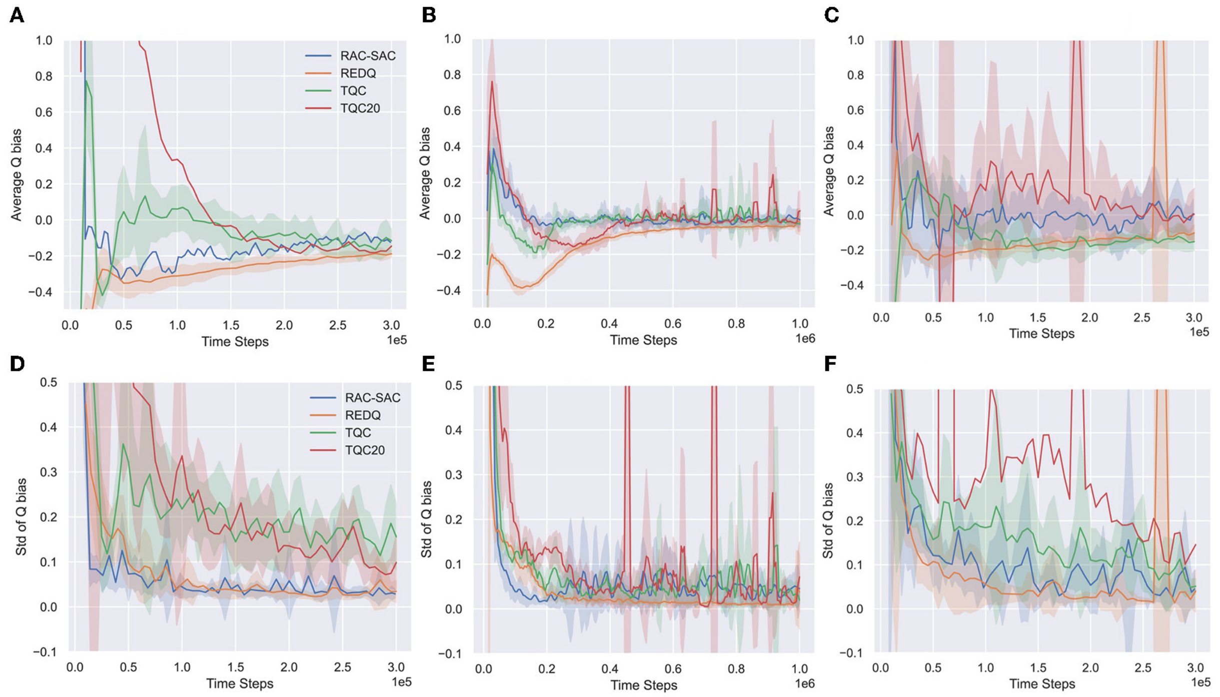


FIGURE 6
Estimated mean and standard deviation of normalized Q bias of RAC-SAC, REDQ, TQC, and TQC20 for Ant and Humanoid with Monte Carlo method. **(A)** Q bias of Ant, **(B)** Q bias of Humanold, **(C)** Q bias of Walker2d, **(D)** Q standard deviation of Ant, **(E)** Q standard deviation of Humanold, and **(F)** Q standard deviation of Walker2d.

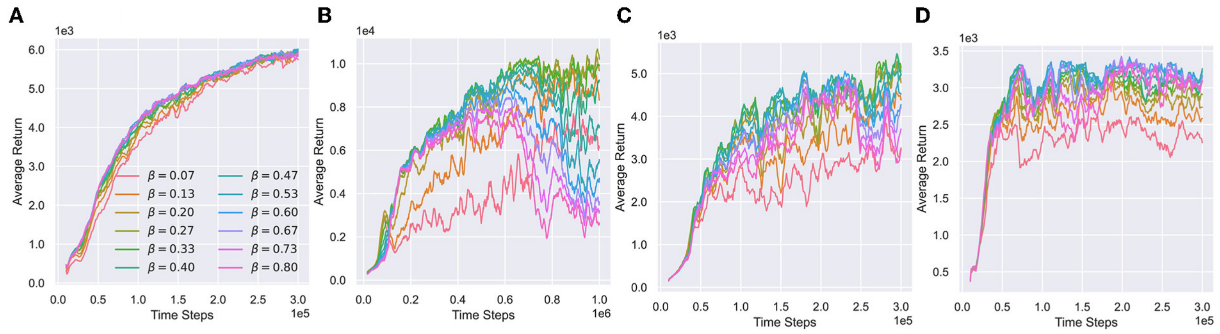


FIGURE 7
Performance of various value confidence bounds with respect to different β. **(A–D)** Performance respect to different β in Ant, Humanoid, Walker2d, and Hopper. We visualize different β belonging to training distribution $U_1 = \mathcal{U}[0, a]$ during training processes.

An underestimated value is enough to guide the policy to learn stably.

6.4. Variants of RAC

We evaluate the performance contributions of ingredients of RAC (punished Bellman backup, policy family, optimistic exploration, independent temperature network, and learning rate warm-up) on a subset of four environments (see Figure 8).

6.4.1. Punished Bellman backup

When using the in-target minimization instead of punished Bellman backup, RAC is stable, but the performance is significantly worse in Humanoid. Punished Bellman backup provides more finer-grained bias control than in-target minimization, reducing the difficulty of learning representations. Compared with other environments, Humanoid has stronger requirements for state representation learning (Chen et al., 2021). Thus, punished Bellman backup far outperforms in-target minimization in Humanoid and is almost the same in other environments.

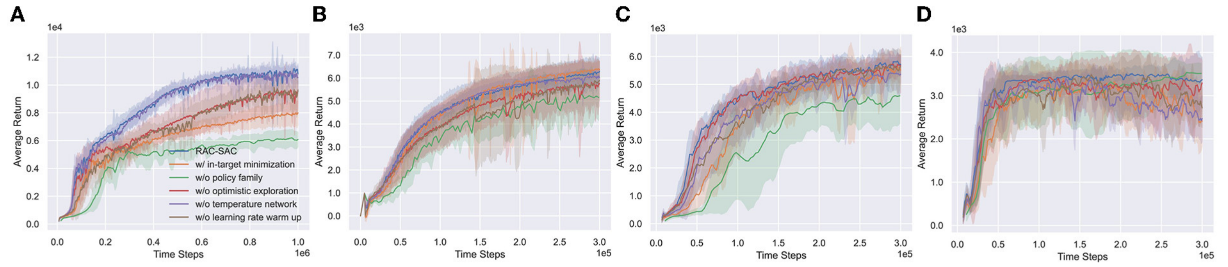


FIGURE 8
Performance of RAC and its variants. **(A–D)** Humanoid, Ant, Walker2d, and Hopper. The in-target minimization version of RAC is shown in Appendix B.4. RAC without policy family is named Vanilla RAC (see Appendix B.3 for more details. In this case, OAC (Ciosek et al., 2019) is used as optimistic exploration method).

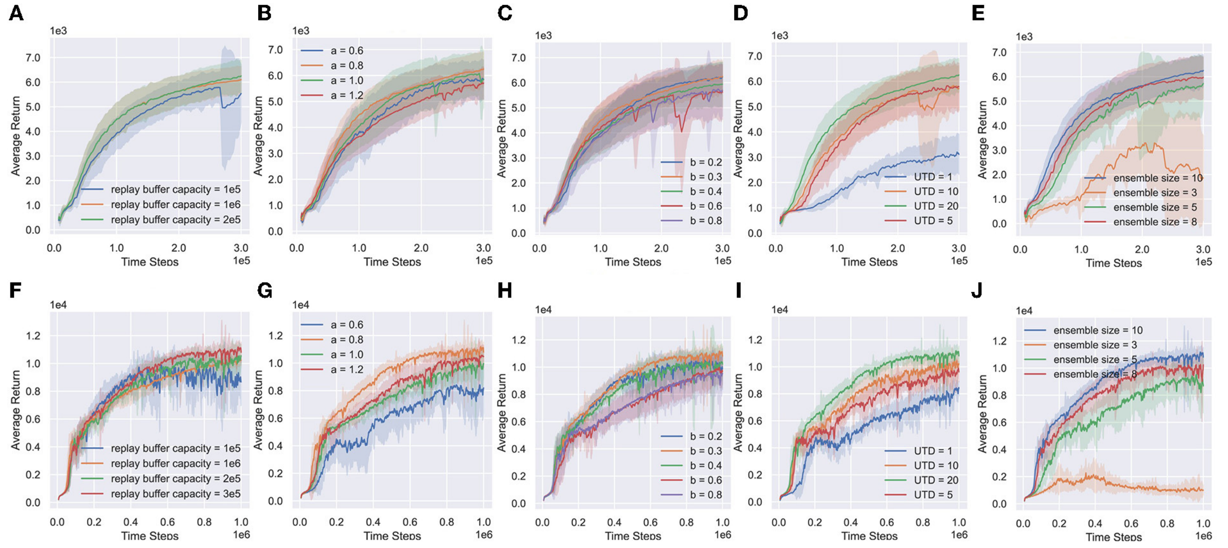


FIGURE 9
Hyperparameter ablations of RAC. **(A–E)** Replay buffer capacity, right side of exploitation distribution (a), right side of exploration distribution (b), the UTD ratio and the ensemble size for ant. **(F–J)** Replay buffer capacity, right side of exploitation distribution (a), right side of exploration distribution (b), the UTD ratio and the Ensemble size for Humanoid.

6.4.2. Policy family

The policy family is paramount to performance. This is consistent with Section 5.3's conjecture. Even with OAC, an agent can only converge to a local optimum without the policy family in Humanoid, indicating that a single optimistic exploration method cannot solve the pessimistic underexploration well. In addition, the convergence speed of the policy has decreased in Walker2d and Ant.

6.4.3. Optimistic exploration

Experimental results support the point in Section 6.3. Optimistic exploration can help the agent escape from local optima in Humanoid. However, in simple environments (like Ant, Walker2d, and Hopper), optimistic exploration has little impact on performance.

6.4.4. Independent temperature network

Except for Walker2d, the independent temperature network has little effect on RAC performance. The learned temperatures are shown in Appendix C. In practice, we find that the independent temperature network can control the entropy of the policy more quickly and stably.

6.4.5. Learning rate warm-up

A high UTD ratio can easily lead to an excessive accumulation of overestimation errors in the early stage of learning. The learning rate warm-up can alleviate this problem and stabilize the learning process. Without the learning rate warm-up, RAC learns slower at the beginning of the training process.

6.5. Hyperparameter ablations

RAC introduces some hyperparameters: (1) replay buffer capacity; (2) right side of exploitation distribution U_1 (a); (3) right side of exploration distribution U_2 (b); (4) UTD ratio G in Algorithm 1; and (5) Ensemble size. Figure 9 shows the numerical results.

Replay buffer capacity (Figures 9A, F). RAC can benefit from a smaller capacity but will be hurt when the capacity is excessively small.

The right side of U_1 (a) (Figures 9B, G). a is a key hyperparameter of RAC. Because a controls the underestimation bias of RAC, which determines the lower bound of Q-functions. The learning process becomes stable with a increasing. However, if a is too large, it will reduce the learning opportunity of optimistic policies, thereby reducing the learning efficiency.

The right side of U_2 (b) (Figures 9C, H). Exploration policies become more conservative with b increasing, and the performance of RAC gradually declines. The increasing standard deviation means that more and more agents fall into local-optimal policies. However, if b is too small, policies may over-explore the overestimated state, resulting in a decrease in learning efficiency.

The ensemble size (Figures 9E, J) and the UTD ratio (Figures 9D, I). RAC appears to benefit greatly from the ensemble size and UTD ratio. When the ensemble size and UTD ratio are increased, we generally get a more stable average bias, a lower standard deviation of bias, and stronger performance.

7. Conclusion

In this study, we empirically discussed under-/overestimation trade-off on improving the sample efficiency in DRL and proposed the Realistic Actor-Critic (RAC), which learns together values and policies with different trade-offs between underestimation and overestimation in the same network. This study proposed Punished Bellman backup that provides fine-granular estimation bias control to make value approximation smoothly shift between upper bounds and lower bounds. This study also discussed the role of the various components of RAC. Experiments show advantageous properties of RAC: low-value approximation error and brilliant sample efficiency. Furthermore, continuous control benchmarks suggest that RAC consistently improves performances and sample efficiency of existing off-policy RL algorithms, such as SAC and TD3. It is of great significance for promoting reinforcement learning in the robot control domain.

Our results suggest that directly incorporating uncertainty to value functions and learning a powerful policy family can provide a promising avenue for improved sample efficiency and performance. Further exploration of ensemble methods, including high-level policies or more rich policy classes, is an exciting avenue for future work.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

SL implemented the code and drafted the manuscript. QT assisted in implementing the code and discussed the manuscript. YP assisted in implementing the code and discussed the manuscript. XM guided the research and discussed the results. GW guided the research, implemented parts of the code, and revised the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This study was funded by the National Natural Science Foundation of Heilongjiang Province (Grant No. YQ2020E028), the National Natural Science Foundation of China (Grant No. 51779059), and in part by the Research Fund from the Science and Technology on Underwater Vehicle Technology under Grant No. 2021-SYSJJ-LB06909.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.1081242/full#supplementary-material

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., et al. (2021). A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Inf. Fusion.* 76, 243–297. doi: 10.1016/j.inffus.2021.05.008

Amos, B., Dinh, L., Cabi, S., Rothörl, T., Colmenarejo, S. G., Muldal, A., et al. (2018). Learning awareness models. *arXiv preprint arXiv:1804.06318.* doi: 10.48550/arXiv.1804.06318

Anschel, O., Baram, N., and Shimkin, N. (2017). "Averaged-DQN: variance reduction and stabilization for deep reinforcement learning," in *International Conference on Machine Learning* (PMLR), 176–185. Available online at: http://proceedings.mlr.press/v70/anschel17a/anschel17a.pdf

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, Z. D., et al. (2020a). "Agent57: outperforming the atari human benchmark," in *International Conference on Machine Learning* (PMLR), 507–517. Available online at: http://proceedings.mlr.press/v119/badia20a/badia20a.pdf

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., et al. (2020b). Never give up: learning directed exploration strategies. *arXiv preprint arXiv:2002.06038.* doi: 10.48550/arXiv.2002.06038

Brafman, R. I., and Tennenholtz, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.* 3, 213–231. doi: 10.1162/153244303765208377

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al. (2016). Openai gym. *arXiv preprint arXiv:1606.01540.* doi: 10.48550/arXiv.1606.01540

Chen, G., and Peng, Y. (2019). Off-policy actor-critic in an ensemble: achieving maximum general entropy and effective environment exploration in deep reinforcement learning. *arXiv preprint arXiv:1902.05551.* doi: 10.48550/arXiv.1902.05551

Chen, L., Jiang, Z., Cheng, L., Knoll, A. C., and Zhou, M. (2022). Deep reinforcement learning based trajectory planning under uncertain constraints. *Front. Neurorobot.* 16, 883562. doi: 10.3389/fnbot.2022.883562

Chen, R. Y., Sidor, S., Abbeel, P., and Schulman, J. (2017). Ucb exploration *via* q-ensembles. *arXiv preprint arXiv:1706.01502.* doi: 10.48550/arXiv.1706.01502

Chen, X., Wang, C., Zhou, Z., and Ross, K. (2021). Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982.* doi: 10.48550/arXiv.2101.05982

Ciosek, K., Vuong, Q., Loftin, R., and Hofmann, K. (2019). "Better exploration with optimistic actor critic," in *Advances in Neural Information Processing Systems 32*. Available online at: https://papers.nips.cc/paper/2019/file/a34bacf839b923770b2c360eefa26748-Paper.pdf

Dorner, F. E. (2021). Measuring progress in deep reinforcement learning sample efficiency. *arXiv preprint arXiv:2102.04881.* doi: 10.48550/arXiv.2102.04881

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., et al. (2020). An empirical investigation of the challenges of real-world reinforcement learning. *arXiv preprint arXiv:2003.11881.* doi: 10.48550/arXiv.2003.11881

Dusenberry, M., Jerfel, G., Wen, Y., Ma, Y., Snoek, J., Heller, K., et al. (2020). "Efficient and scalable bayesian neural nets with rank-1 factors," in *International Conference on Machine Learning* (PMLR), 2782–2792. Available online at: http://proceedings.mlr.press/v119/dusenberry20a/dusenberry20a.pdf

Fujimoto, S., Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning* (PMLR), 1587–1596. Available online at: http://proceedings.mlr.press/v80/fujimoto18a/fujimoto18a.pdf

Fukuda, T. (2020). Cyborg and bionic systems: Signposting the future. *Cyborg Bionic Syst.* 2020, 1310389. doi: 10.34133/2020/1310389

Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (JMLR Workshop and Conference Proceedings), 315–323. Available online at: http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

Goyal, A., Sodhani, S., Binas, J., Peng, X. B., Levine, S., and Bengio, Y. (2019). Reinforcement learning with competitive ensembles of information-constrained primitives. *arXiv preprint arXiv:1906.10667.* doi: 10.48550/arXiv.1906.10667

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., et al. (2018). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905.* doi: 10.48550/arXiv.1812.05905

Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., et al. (2020). Training independent subnetworks for robust prediction. *arXiv preprint arXiv:2010.06610.* doi: 10.48550/arXiv.2010.06610

He, Q., Gong, C., Qu, Y., Chen, X., Hou, X., and Liu, Y. (2021). MEPG: a minimalist ensemble policy gradient framework for deep reinforcement learning. *arXiv preprint arXiv:2109.10552.* doi: 10.48550/arXiv.2109.10552

Janner, M., Fu, J., Zhang, M., and Levine, S. (2019). "When to trust your model: Model-based policy optimization," in *Advances in Neural Information Processing Systems 32*. Available online at: https://dl.acm.org/doi/10.5555/3454287.3455409

Jung, W., Park, G., and Sung, Y. (2020). Population-guided parallel policy search for reinforcement learning. *arXiv preprint arXiv:2001.02907.* doi: 10.48550/arXiv.2001.02907

Kalweit, G., and Boedecker, J. (2017). "Uncertainty-driven imagination for continuous deep reinforcement learning," in *Conference on Robot Learning* (PMLR), 195–206. Available online at: http://proceedings.mlr.press/v78/kalweit17a/kalweit17a.pdf

Karimpanal, T. G., and Bouffanais, R. (2018). Experience replay using transition sequences. *Front. Neurorobot.* 12, 32. doi: 10.3389/fnbot.2018.00032

Kim, H., Kim, J., Jeong, Y., Levine, S., and Song, H. O. (2019). "EMI: exploration with mutual information," in *International Conference on Machine Learning* (PMLR), 3360–3369. Available online at: https://arxiv.org/pdf/1810.01176.pdf

Kingma, D. P., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* doi: 10.48550/arXiv.1412.6980

Kumar, A., Gupta, A., and Levine, S. (2020). Discor: Corrective feedback in reinforcement learning *via* distribution correction. *Adv. Neural Inf. Process. Syst.* 33, 18560–18572. doi: 10.48550/arXiv.2003.07305

Kuznetsov, A., Grishin, A., Tsypin, A., Ashukha, A., and Vetrov, D. (2021). Automating control of overestimation bias for continuous reinforcement learning. *arXiv preprint arXiv:2110.13523.* doi: 10.48550/arXiv.2110.13523 Available online at: https://arxiv.org/pdf/2110.13523.pdf

Kuznetsov, A., Shvechikov, P., Grishin, A., and Vetrov, D. (2020). "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *International Conference on Machine Learning* (PMLR), 5556–5566.

Lan, Q., Pan, Y., Fyshe, A., and White, M. (2020). Maxmin q-learning: controlling the estimation bias of q-learning. *arXiv preprint arXiv:2002.06487.* doi: 10.48550/arXiv.2002.06487

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. (2021). "Sunrise: a simple unified framework for ensemble learning in deep reinforcement learning," in *International Conference on Machine Learning* (PMLR), 6131–6141. Available online at: http://proceedings.mlr.press/v139/lee21g/lee21g.pdf

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Rob. Res.* 37, 421–436. doi: 10.1177/0278364917710318

Lyle, C., Rowland, M., Ostrovski, G., and Dabney, W. (2021). "On the effect of auxiliary tasks on representation dynamics," in *International Conference on Artificial Intelligence and Statistics* (PMLR), 1–9. Available online at: http://proceedings.mlr.press/v130/lyle21a/lyle21a.pdf

Namiki, A., and Yokosawa, S. (2021). Origami folding by multifingered hands with motion primitives. *Cyborg Bionic Syst.* 2021, 9851834. doi: 10.34133/2021/9851834

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). "Deep exploration *via* bootstrapped DQN," in *Advances in Neural Information Processing Systems 29*. Available online at: https://papers.nips.cc/paper/2016/file/8d8818c8e140c64c743113f563cf750f-Paper.pdf

Parker-Holder, J., Pacchiano, A., Choromanski, K. M., and Roberts, S. J. (2020). Effective diversity in population based reinforcement learning. *Adv. Neural Inf. Process. Syst.* 33, 18050–18062. doi: 10.48550/arXiv.2002.00632

Pathak, D., Gandhi, D., and Gupta, A. (2019). "Self-supervised exploration *via* disagreement," in *International Conference on Machine Learning* (PMLR), 5062–5071. Available online at: http://proceedings.mlr.press/v97/pathak19a/pathak19a.pdf

Peer, O., Tessler, C., Merlis, N., and Meir, R. (2021). Ensemble bootstrapping for q-learning. *arXiv preprint arXiv:2103.00445.* doi: 10.48550/arXiv.2103.00445

Pendrith, M. D., and Ryan, M. R. (1997). *Estimator variance in reinforcement learning: Theoretical problems and practical solutions.* University of New South Wales, School of Computer Science and Engineering.

Rashid, T., Peng, B., Böhmer, W., and Whiteson, S. (2020). "Optimistic exploration even with a pessimistic initialization," in *International Conference on Learning Representations (ICLR)*. doi: 10.48550/arXiv.2002.12174

Saphal, R., Ravindran, B., Mudigere, D., Avancha, S., and Kaul, B. (2020). SEERL: sample efficient ensemble reinforcement learning. *arXiv preprint arXiv:2001.05209*. doi: 10.48550/arXiv.2001.05209

Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). "Universal value function approximators," in *International Conference on Machine Learning* (PMLR), 1312–1320. Available online at: http://proceedings.mlr.press/v37/schaul15.pdf

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press. Available online at: http://www.incompleteideas.net/sutton/book/first/Chap1PrePub.pdf

Thrun, S., and Schwartz, A. (1993). "Issues in using function approximation for reinforcement learning," in *Proceedings of the Fourth Connectionist Models Summer School* (Hillsdale, NJ), 255–263.

Todorov, E., Erez, T., and Tassa, Y. (2012). "MuJoCo: a physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura-Algarve: IEEE), 5026–5033.

Van Hasselt, H., Guez, A., and Silver, D. (2016). "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30*. Available online at: https://ojs.aaai.org/index.php/AAAI/article/download/10295/10154

Warwick, D. P., and Lininger, C. A. (1975). *The Sample Survey: Theory and Practice*. McGraw-Hill. Available online at: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=The+Sample+Survey%3A+Theory+and+Practice&btnG=

Wen, Y., Tran, D., and Ba, J. (2020). Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*. doi: 10.48550/arXiv.2002.06715

Wenzel, F., Snoek, J., Tran, D., and Jenatton, R. (2020). Hyperparameter ensembles for robustness and uncertainty quantification. *Adv. Neural Inf. Process. Syst*. 33, 6514–6527. doi: 10.48550/arXiv.2006.13570

Wu, Y., Zhai, S., Srivastava, N., Susskind, J., Zhang, J., Salakhutdinov, R., et al. (2021). Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*. doi: 10.48550/arXiv.2105.08140

Yu, Y. (2018). "Towards sample efficient reinforcement learning," in *IJCAI*, 5739–5743. Available online at: https://www.ijcai.org/proceedings/2018/0820.pdf

Zheng, Z., Yuan, C., Lin, Z., and Cheng, Y. (2018). "Self-adaptive double bootstrapped DDPG," in *International Joint Conference on Artificial Intelligence*. Available online at: https://www.ijcai.org/proceedings/2018/0444.pdf

Ziebart, B. D. (2010). *Modeling Purposeful Adaptive Behavior With the Principle of Maximum Causal Entropy*. Carnegie Mellon University. Available online at: http://reports-archive.adm.cs.cmu.edu/anon/anon/home/ftp/usr/ftp/ml2010/CMU-ML-10-110.pdf

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 08 February 2023
DOI 10.3389/fnbot.2022.1074862

Check for updates

OPEN ACCESS

EDITED BY
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

REVIEWED BY
Chi-Man Vong,
University of Macau, China
Ning Sun,
Nankai University, China

*CORRESPONDENCE
Wenjing Pei
✉ wenjingpei@buaa.edu.cn

RECEIVED 20 October 2022
ACCEPTED 28 December 2022
PUBLISHED 08 February 2023

CITATION
Pei W, Shi Z and Gong K (2023) Small target detection with remote sensing images based on an improved YOLOv5 algorithm. *Front. Neurorobot.* 16:1074862. doi: 10.3389/fnbot.2022.1074862

Small target detection with remote sensing images based on an improved YOLOv5 algorithm

Wenjing Pei[1]*, Zhanhao Shi[2] and Kai Gong[1]

[1]The Seventh Research Division and the Center for Information and Control, School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing, China, [2]School of Information Science and Engineering, Shandong Agriculture and Engineering University, Jinan, China

Introduction: Small target detection with remote sensing images is a challenging topic due to the small size of the targets, complex, and fuzzy backgrounds.

Methods: In this study, a new detection algorithm is proposed based on the YOLOv5s algorithm for small target detection. The data enhancement strategy based on the mosaic operation is applied to expand the remote image training sets so as to diversify the datasets. First, the lightweight and stable feature extraction module (LSM) and C3 modules are combined to form the feature extraction module, called as LCB module, to extract more features in the remote sensing images. Multi-scale feature fusion is realized based on the Res 2 unit, Dres 2, and Spatial Pyramid Pooling Small (SPPS) models, so that the receptive field can be increased to obtain more multi-scale global information based on Dres2 and retain the obtained feature information of the small targets accordingly. Furthermore, the input size and output size of the network are increased and set in different scales considering the relatively less target features in the remote images. Besides, the Efficient Intersection over Union (EIoU) loss is used as the loss function to increase the training convergence velocity of the model and improve the accurate regression of the model.

Results and discussion: The DIOR-VAS and Visdrone2019 datasets are selected in the experiments, while the ablation and comparison experiments are performed with five popular target detection algorithms to verify the effectiveness of the proposed small target detection method.

KEYWORDS

small target detection, remote sensing images, YOLOv5s, deep learning, EIoU loss

1. Introduction

With the development of remote sensing technologies, a large amount of remote sensing images can be obtained from video satellites and unmanned aerial vehicles (UAVs) (Hu et al., 2019; Zhang et al., 2019; Hou et al., 2020; Lu et al., 2020; Wang et al., 2020; Pei and Lu, 2022). Recently, remote sensing image processing has attracted widespread attention, such as target detection, classification, tracking, and surveillance (Jia, 2000, 2003; Guo et al., 2017; Wang et al., 2018; Yin et al., 2020; Zhong et al., 2020; Jiang et al., 2021; Dong et al., 2022; Habibzadeh et al., 2022; Ma and Wang, 2022; Pei, 2022). Particularly, target detection is a hot topic with remote sensing images (TDRSIs), where the TDRSI has been widely applied in the fields of military, transportation, forest survey, security monitoring, disaster monitoring, and so on (Zhang et al., 2016; Han et al., 2017; Zhu et al., 2017). Therefore, TDRSI is a significant and challenging task due to the small size of the targets, high speed detection, and high accuracy requirements (Zhang et al., 2017; Dong et al., 2022).

Target detection aims to find all interested objects in the images, which has been studied with the development of computer vision technologies in recent decades. Numerous algorithms, especially convolutional neural networks (CNNs), have been widely employed for general target detection, such as SSD, YOLO, R-CNN, and Faster R-CNN (He et al., 2016; Li et al., 2019, 2021; Zhong et al., 2020; Fan et al., 2021; Tu et al., 2021; Dong et al., 2022; Mikriukov et al., 2022). For instance, Lawal (2021) have proposed a modified YOLOv3 model to detect tomatoes in complex environments. Wu et al. (2018) presented a different scaled algorithm based on the Faster R-CNN to solve small-scaled face detection. YOLOv3 network can be used for blood cell recognition (Shakarami et al., 2021) while a YOLOv4 algorithm can be used for oil well detection (Shi et al., 2021).

Considering general target detection, small target detection in remote sensing images is more difficult due to several reasons (refer to Figure 1) (Meng, 2012; Li, 2016; Du et al., 2018; Chen et al., 2021; Liu et al., 2022). First, the scales of the remote sensing images may be relatively large compared to the small target size or clustered targets in the images. Moreover, the background of the remote sensing images could be complex and fuzzy, sometimes even similar to the target features. Third, there is not enough feature information of the targets in one image, i.e., vehicles, pedestrians, and others have only few pixels for object detection in the optical remote sensing images (DIOR) (Li et al., 2020) and Visdrone2019 (Zhu et al., 2019) datasets.

Hence, a lot of methods have been developed specifically to achieve small target detection in remote sensing images. For instance, Lu et al. (2021) have proposed a single shot detection (SSD) to detect the small target with complex background and scale variations. An improved YOLOv3 model has been designed for ship detection in remote sensing images with high accuracy and robustness (Xu, 2020). In Wang J. et al. (2020), an end-to-end feature-reflowing pyramid network has been proposed for multi-scale and multi-class object detection. Furthermore, a novel cascaded rotating-anchor-assisted detection network has been presented in Yu et al. (2022) to improve ship detection accuracy with aerial images. Moreover, Huang et al. designed a lightweight target detector to rapidly and accurately detect small targets (Huang et al., 2022). A detection algorithm based on the feature balancing and refinement network is developed to successfully detect ships (Fu et al., 2020). A squeeze-and-excitation YOLOv3 algorithm has been designed for small target detection in remote sensing images with low computation costs (Zhou et al., 2021). Moreover, Ling et al. (2022) have developed a new time-delay feedback model to detect small target motion in complex dynamic backgrounds. An indoor small target detection algorithm is described in Huang L. et al. (2022) based on multi-scale feature fusion to improve the accuracy and speed of the target detection.

Based on the above analysis, this study presents an improved LCB-YOLOv5s detection algorithm for remote sensing images. First, a new module comprising the lightweight and stable module (LSM) and cross-stage partial networks with three convolutions (C3) structure module where these modules are combined to form the feature extraction module, called as LCB module, is designed to extract numerous features of small targets. Then, the Spatial Pyramid Pooling Small (SPPS) module is developed to increase the weight of these features in the spatial dimension. Moreover, the Duble Res2Net (Dres2) module is used in the head to increase the receptive field so as to obtain more multi-scale global information and realize fine-grained feature extraction. In order to overcome the difficulty of relatively few features, the input size of the network is increased with different output feature map sizes. In summary, the contributions of the paper are summarized as follows:

1) An LCB-YOLOv5 algorithm has been developed for small target detection with remote sensing images. In the feature extraction module, the LCB module is configured based on the LSM and C3 modules to extract more features. Moreover, the SPPS and Dres2 modules are introduced to increase the weight of the features in the receptive field and so as to extract more multi-scale global information.
2) In order to improve the accuracy of the small target detection, the input size of the network is increased from 640 × 640 to 1,024 × 1,024, and the output feature map size is set as 32 × 32, 64 × 64, and 128 × 128, respectively.
3) The *EIoU* function is employed as the loss function to increase the training convergence velocity of the model and the regression accuracy for the target detection.

The remainder of the paper is organized as follows. Section 2 describes the proposed method in detail. Experiments of the small target detection with the selected datasets are performed and the results are analyzed in Section 3. The conclusion is provided in Section 4.

2. The proposed method

This section presents the details of the proposed method. As shown in Figure 2, except for the first layer, the 3 × 3 convolutional layers in the backbone of the YOLOv5s detection algorithm are replaced with the LSM module. Since small targets have fewer features than those large targets in the images, the SPPS module is designed to increase the weight of these features in the spatial dimension. The Dres2 module is further introduced in the head with the strategy of multi-scale feature fusion to enhance the small target detection performance. The input size of the network is also increased with

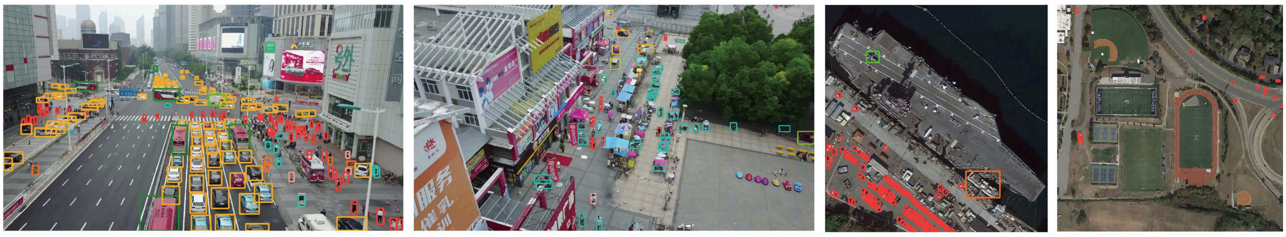


FIGURE 1
Examples of targets in remote sensing images.

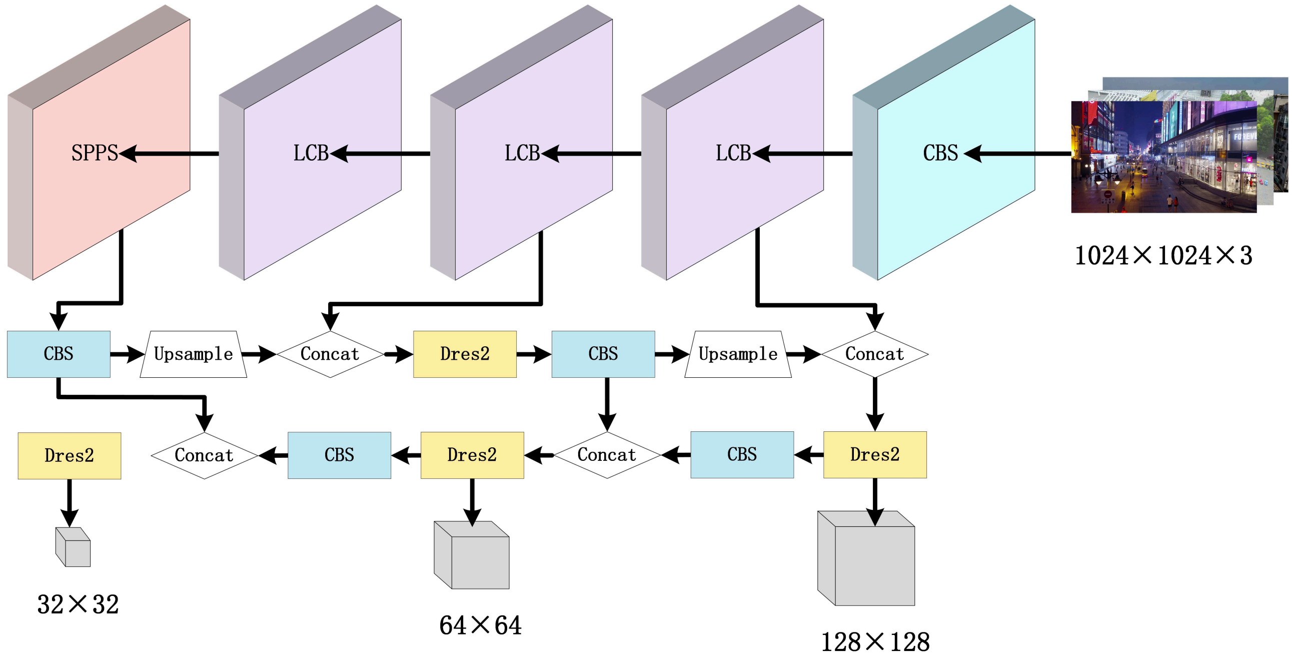


FIGURE 2
The architecture of the developed LCB-YOLOv5s.

various output feature map sizes, while the EIoU loss function is designed to increase the convergence speed.

2.1. Data augmentation

In general, the original training data have to be pre-processed to meet the training requirement; hence, many data enhancement strategies are employed to expand and diversify the remote sensing images so as to improve the generalization ability of the trained model and to minimize the irrelevant characteristic information in the training data. As shown in Figure 3, the mosaic operation is applied to enrich the datasets, where four original images can be randomly selected from a batch in the datasets to perform a flip, translate, change the color gamut, and stitch the images such operations. Based on the above data enhancement operations, the size of the images is relatively close to the small targets, and the number of small targets can be increased in the remote sensing images. Therefore, the small target datasets can be expanded, which can effectively improve the small target detection ability of the model. Accordingly, the demand for GPU memory can be reduced and the training speed can be also improved greatly.

2.2. Feature extraction module

In the remote sensing images, the sizes of the targets may be small and the edges of the targets may be blurred. Hence, a LCB feature extraction module is designed to improve the target detection performance, as shown in Figure 4. Specifically, numerous features of the small targets can be extracted using the LSM module. The standard 3×3 convolution is used for feature extraction, and some significant features of the original feature map are preserved using maximum pooling. Then, the output feature map is enriched by concatenation. Moreover, the C3 module can perform feature extraction and fusion, where 1 × 1 convolution is applied to reduce the dimension of the original feature map, and the feature map after convolutional extraction is spliced as the output.

It is known that the *conv* + *batch normalization* + *silu* (CBS) and *conv* + *batch normalization* + *relu* (CBR) modules are two types of standard convolution modules. As shown in Figures 5A, B, CBS and CBR utilize the convolution operation, batch normalization (BN), and activation function, where the SiLU and the ReLU are employed as the activation functions, respectively. It is noted that the CBR module with the ReLU can reduce the amount of calculation and eliminate the gradient diminishing, where the activating function with ReLu can learn faster than the sigmoid or tanh functions.

Figure 6 displays the proposed LSM module, mainly composed of convolution and pooling branches. First, the 1 × 1 standard convolution and 3 × 3 convolution are used to reduce the data dimension and extract features, respectively. Then, the 1 × 1 standard convolution is used once again to increase the data dimension. Furthermore, the feature map is subsampled by 2 × 2 max pooling and the number of channels is adjusted based on the 1 × 1 standard convolution. Finally, the output is obtained based on the Concat module with the above features. Compared to the traditional 3 × 3 convolution, LSM can obtain more abundant features. On the other hand, LSM can preserve some significant features of the original feature map based on the maximum pooling. On the other hand, LSM can enrich the feature map and merge it as the output.

The Res unit is a standard residual module, which is depicted in Figure 7. The 1 × 1 standard convolution is used to reduce

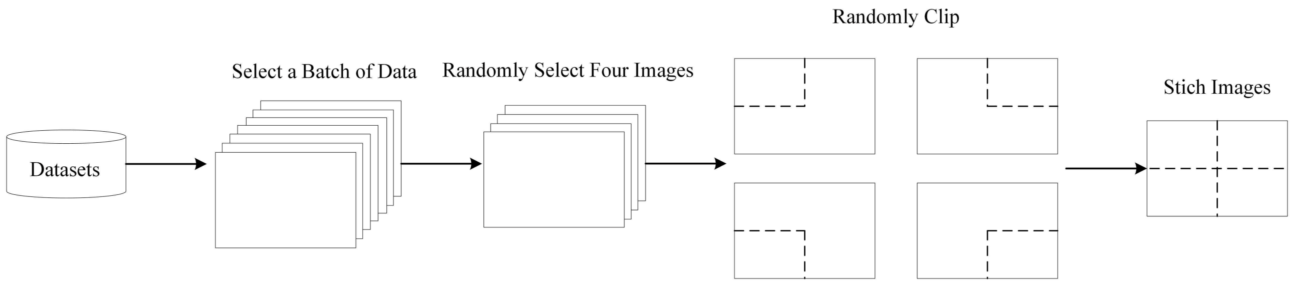


FIGURE 3
The procedure of the mosaic data enhancement.

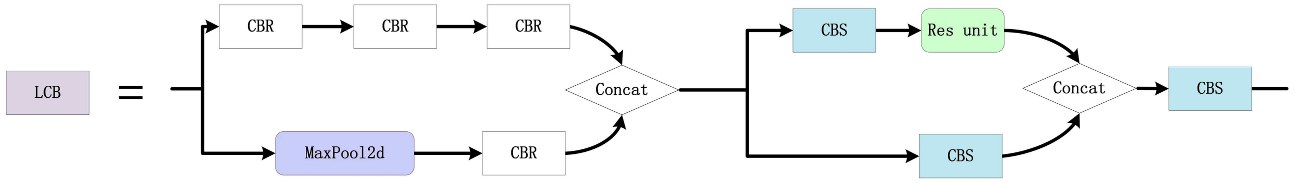


FIGURE 4
The structure of the LCB module.

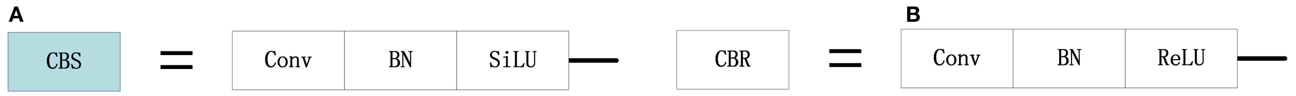


FIGURE 5
The schematic diagram of the standard convolution: **(A)** CBS module, **(B)** CBR module.

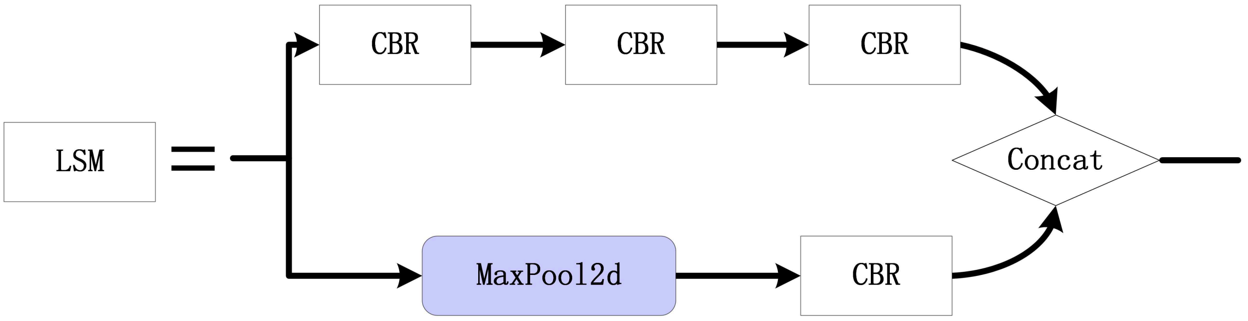


FIGURE 6
LSM module configuration.

the dimension, and the 3 × 3 convolution is used to extract features. Then, the original information and feature information after convolution are added as the output. The C3 module is used for feature extraction and feature fusion, as described in Figure 8. Hence, the rich semantic information and features are obtained to convolve the upper layer feature map based on the Res unit and the 3 × 3 convolution is applied to extract features. Then, 1 × 1 convolution is applied to reduce the dimension of the original feature map, which is spliced with the convolved feature maps and used as the output.

2.3. Feature fusion module

In order to improve the accuracy of the small target detection, the Res2 unit module is designed (refer to Figure 9), where multigroup 3 × 3 convolutions are cascaded to enlarge the receptive field of the network and the features of each group are fused. The Dres2 module is further designed based on the C3 module (refer to Figure 10), where the original residual block is replaced with two Res2 modules. Compared to the C3 module, the Dres2 module can increase the receptive field to obtain more multi-scale global

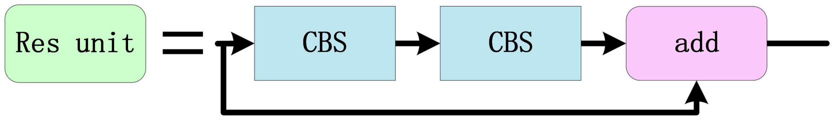


FIGURE 7
Res unit module.

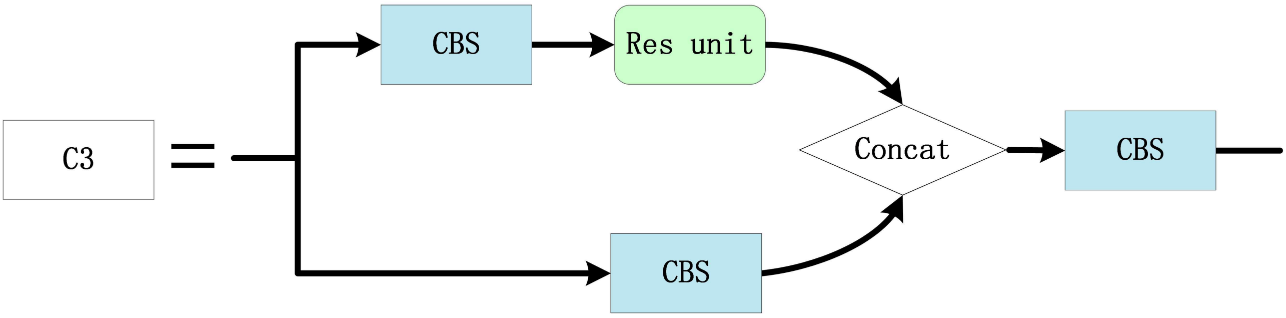


FIGURE 8
C3 module configuration.

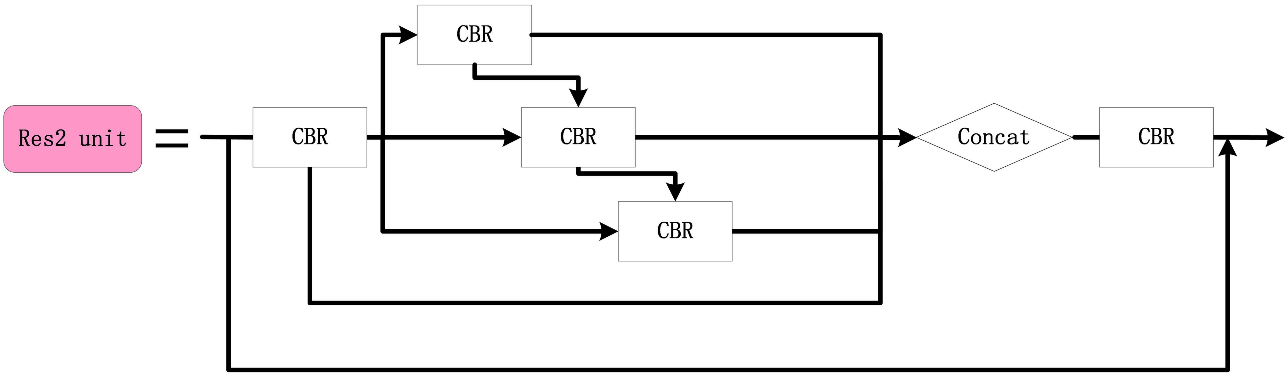


FIGURE 9
The configuration of the Res2 unit module.

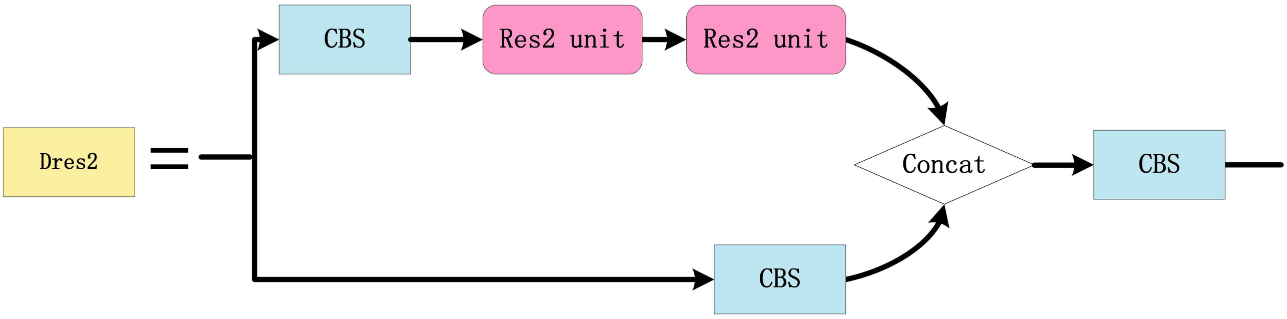


FIGURE 10
The configuration of the Dres2 module.

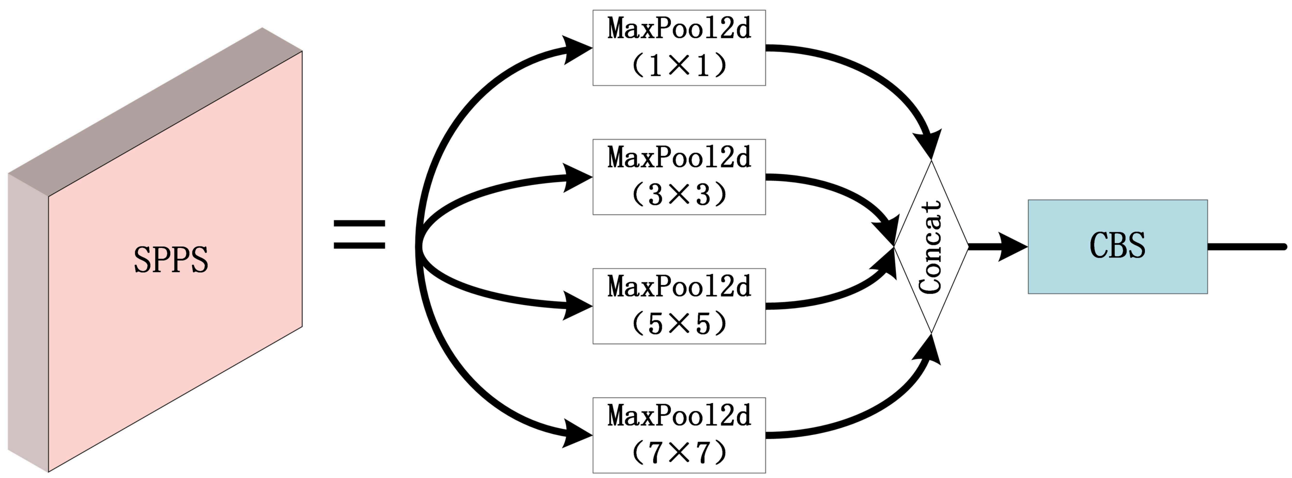


FIGURE 11
The configuration of the SPPS module.

information. Therefore, the Dres2 module is applied here to realize fine-grained feature extraction.

As depicted in Figure 11, the SPPS module is a modified version of the Spatial Pyramid Pooling (SPP) module in the network, where the three groups of maximum pooling are 1 × 1, 3×3, 5 × 5, and 7×7. Since small targets have a relatively small proportion of pixels in the remote sensing images, the effective feature information may be difficult to extract. In order to overcome the above difficulty, the SPPS module applies different sizes of the max pooling kernels, and thus, the feature information of the small targets can be retained accordingly since SPPS not only has the advantages of SPP but also can improve the detection performance for small targets.

2.4. Input size of the network

The input image size of the YOLOv5 network is 640 × 640 and the output size is 80, 40, and 20 in the prediction head. Compared to the YOLOv5 algorithm, the input size of the network and the predicted feature map are maximized to 1,024 and 256, and 128 and 64, respectively. Consequently, the input size of the network is enlarged to overcome the limitation of less small target features in the remote sensing images.

2.5. Loss function

Here, the IoU and GIoU Loss functions of the original YOLOv5 algorithm are first presented to analyze the deficiencies in small target detection. Then, the EIoU Loss is introduced (Zhang et al., 2021), where the GIoU Loss function refers to an improved intersection-over-union (IoU). The IoU is used to denote the intersection ratio of the prediction box (*PB*) and ground truth box (*GB*), which is described as follows:

$$IoU = \frac{PB \cap GB}{PB \cup GB}, \tag{1}$$

Moreover, the *IoU* Loss function is calculated as follows:

$$L_{IoU} = 1 - \frac{PB \cap GB}{PB \cup GB}. \tag{2}$$

However, if there is no intersection between *PB* and *GB*, *IoU* Loss is nearly zero, which can hardly be used to reflect their distance. Moreover, the IoU Loss has a relatively slow convergence rate; hence, the *GIoU* is introduced to avoid such a problem, calculated as follows:

$$GIoU = IoU - \frac{A_c - U}{A_c}, \tag{3}$$

where A_c is the area of the smallest rectangular box including both *PB* and *GB* simultaneously and *U* is the union of *PB* and *GB*. Furthermore, the GIoU Loss can be expressed as follows:

$$L_{GIoU} = 1 - GIoU = 1 - IoU + \frac{A_c - U}{A_c}. \tag{4}$$

It is noted that *GIoU* Loss can be optimized for situations where the *PB* and *GB* are not overlapped. Nevertheless, if these two boxes are positioned relatively close, the values of the *GIoU* and *IoU* Loss are also approximately equal. In order to solve the above problem, the *EIoU* Loss is used as the loss function of LCB-YOLOv5. The *EIoU* and the *EIoU* loss functions are calculated as follows:

$$EIoU = IOU - \frac{\rho^2\left(b, b^{gt}\right)}{c^2} - \frac{\rho^2\left(w, w^{gt}\right)}{c_w^2} - \frac{\rho^2\left(h, h^{gt}\right)}{c_h^2}, \tag{5a}$$

$$L_{EIoU} = L_{IoU} + L_{dis} + L_{asp} = 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(w, w^{gt}\right)}{c_w^2} + \frac{\rho^2\left(h, h^{gt}\right)}{c_h^2}, \tag{5b}$$

where c_w and c_h are the minimum widths and heights of the outer box covering two boxes, respectively. Compared with *IoU* and *GIoU* Loss functions, the distance between the target and anchor, the overlap rate and penalty items are considered based on the *EIoU* Loss function. Therefore, the regression accuracy for detection is more stable and the training convergence speed is faster.

3. Experimental results and analysis

3.1. Experimental settings

The proposed LCB-YOLOv5s network is trained with the RTX 3090, 24G memory, and Ubuntu 20.04.4 operating system, while the proposed network and the comparison algorithms are programmed in Python 3.8 and Cuda 11.3. The hyperparametric configuration is displayed in Table 1. In total, two datasets are selected for the experiments. The first is the VisDrone2019 dataset, which was collected by the Aiskyeye team in the Machine Learning and Data Mining Laboratory of Tianjin University. It includes 10 categories comprising more than 2.6 million annotation boxes. The targets in the VisDrone2019 dataset are pedestrians, people, bicycles, cars, vans, trucks, tricycles, awning-tricycles, buses, and motors. Moreover, the training and validation sets contain 6,471 and 548 remote sensing images, respectively. The other dataset is the DIOR remote sensing dataset, which contains 20 categories with 23,463 remote sensing images and 192,472 examples.

In the experiments, vehicles, ships, and airplanes are selected as the targets from 1,673 remote sensing images. Furthermore, a new dataset called the DIOR-VAS dataset is reconfigured including three types of targets: vehicles, airplanes, and ships. As shown in Table 2, the training and verification sets contain 1,334 and 339 remote sensing images, respectively.

3.2. Evaluation metrics of the experiments

During the experiments, three common evaluation metrics are used to evaluate the effect of the proposed method, mean average precision (*mAP*), precision (*P*), and recall (*R*). Specifically, *P* and *R* are calculated as follows:

$$P = \frac{TruePositives}{TruePositives + FalsePositives}, \tag{6a}$$

$$R = \frac{TruePositives}{TruePositives + \ FalseNegatives}, \tag{6b}$$

where *TruePositives* denotes the targets correctly classified as positive examples, *FalsePositives* denotes the targets incorrectly

TABLE 1 Hyperparametric configuration of the experiments.

Hyperparametric	Epochs	Batch size	Learning rate	Momentum	Weight decay
Configuration	150	16	0.01	0.973	0.0005

TABLE 2 Details of the VisDrone2019 and DIOR datasets.

Datasets	Categories	Totaling images	Training set	Validation set
VisDrone2019	10	8,629	6,471	548
DIOR	20	23,463	5,862	5,863
DIOR-VAS	3	1,673	1,334	339

TABLE 3 Comparison of the proposed method and other approaches based on the Visdrone2019 dataset.

Models	*P* (%)	*R* (%)	*mAP* (%)	Car	Bus	Pedestrian
YOLOv5	42.2	31.5	30.5	0.72	0.38	0.39
PicoDet	35.7	30.5	28.2	0.75	0.33	0.38
YOLOv3	40.5	26.8	25.9	0.65	0.28	0.32
YOLOv3-SPP	42.5	25.1	25.4	0.65	0.26	0.32
YOLOv7	39.5	30.3	26.2	0.72	0.33	0.34
LCB-YOLOv5s	56.2	46.7	47.9	0.86	0.65	0.59

TABLE 4 Comparison of the proposed method with other approaches based on the DIOR-VAS dataset.

Models	*P* (%)	*R* (%)	*mAP* (%)	Vehicle	Airplane	Ship
YOLOv5	93.3	85.8	90.4	0.75	0.99	0.96
PicoDet	81.6	29.3	55.9	0.53	0.54	0.59
YOLOv3	92.7	84	88.6	0.74	0.99	0.93
YOLOv3-SPP	92.9	83.9	88.6	0.74	0.98	0.94
YOLOv7	92.5	85.8	90	0.74	0.99	0.96
LCB-YOLOv5s	93.4	88.6	93	0.84	0.99	0.96

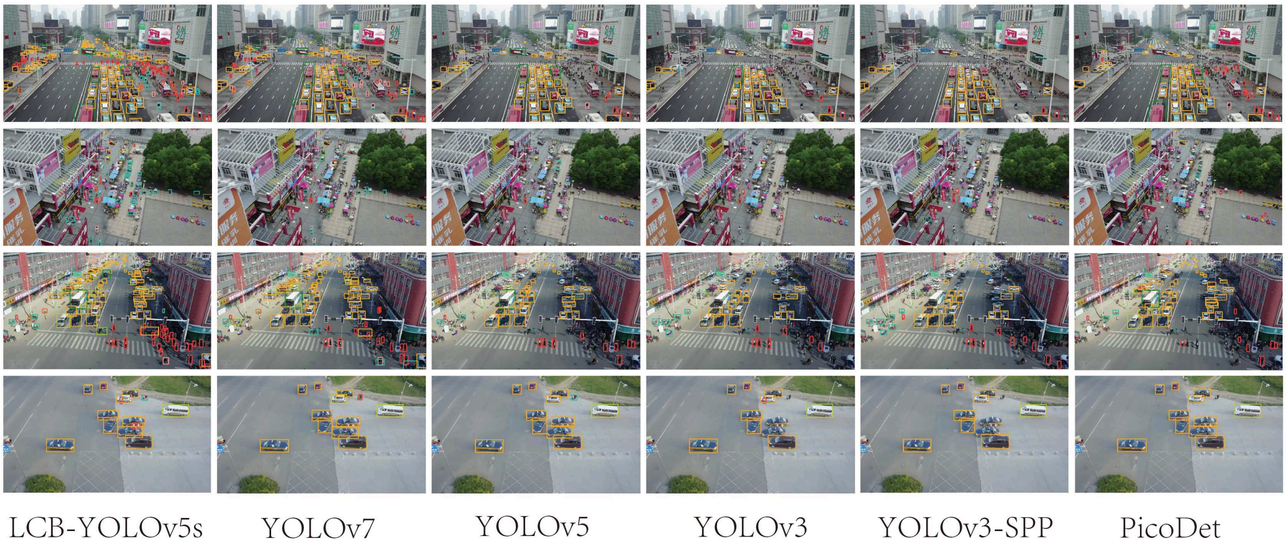


FIGURE 12
Comparison of the target detection of six different models on the Visdrone2019 dataset.

classified as positive examples, and *FalseNegatives* denotes the targets incorrectly classified as negative examples.

Additionally, *AP* is the average classification accuracy of a category in the datasets. It is calculated as follows:

$$AP = \int_0^1 P(R)\,dt \tag{7}$$

where $P(R)$ is the P–R curve to be used to calculate the *AP*. Based on the *AP*, the *mAP* can be obtained as follows:

$$mAP = \frac{\sum_{n=0}^{N} AP_n}{N} \tag{8}$$

where N is the number of the detected target categories.

3.3. Experimental results and analysis

Table 3 displays the comparison results of our proposed method with the other five approaches, *Mets* = {YOLOv5, YOLOv3, YOLOv3-SPP, YOLOv7, PicoDet}, on the Visdrone2019 dataset. The proposed method has achieved significantly higher performance than the other methods, with *P*, *R*, and *mAP* as 56.2, 46.7, and 47.9, respectively. Particularly, the *mAP* of the proposed method is 17.4, 19.7, 22, 22.5, and 21.7 higher than those of the methods in *Mets* one by one. Furthermore, the *P* of the LCB-YOLOv5s is higher by {14, 20.5, 15.7, 13.7, 16.7} in comparison to those of methods in *Mets*. Moreover, the *R* of the LCB-YOLOv5s is higher by {15.2, 16.2, 19.9, 21.6, 16.4} than those of the methods in *Mets* in turn. However, the PicoDet method has a relatively weaker performance in the DIOR-VAS dataset. Furthermore, in Table 3, the LCB-YOLOv5s exhibits much better detection performance than the other five methods for bus and pedestrian detection and slightly better detection performance than the rest methods for plane and ship detection. In general, LCB-YOLOv5s can achieve higher small target detection performance with a reduced false detection rate.

Table 4 illustrates the comparison results of the proposed method with the other five methods on the DIOR-VAS dataset, where vehicles, airplanes, and ships are selected as the small targets. The proposed method exhibits a better performance than the other methods, with *mAP*, *P*, and *R* of 93, 93.4, and 88.6, respectively. Particularly, the *mAP*, *P*, and *R* of YOLOv5s and YOLOv7 are 90.4, 93.3, and 85.8 and 90, 92.5, and 85.8, respectively. Thus, the *mPA* and *R* of the YOLOv3 and YOLOv3-SPP are lower by 4.4, 4.6, 4.4, and 4.7, respectively. The *R* of the YOLOv3 and YOLOv3-SPP is also relatively lower. Figure 12 displays the target detection results of the six algorithms on the Visdrone2019 dataset, where the orange, green, and red boxes indicate the detected targets of cars, buses, and pedestrians, respectively. Compared to the other five algorithms, LCB-YOLOv5s can accurately detect more targets, especially buses and pedestrians, although the prediction boxes are densely distributed in the leftmost subfigure of Figure 12. This demonstrates that the proposed LCB-YOLOv5 algorithm has an advantage over the other algorithms for small target detection. The target detection comparison of the six algorithms on the DIOR-VAS dataset is illustrated in Figure 13, where the orange, green, and red boxes are the detection results of the ships, airplanes, and vehicles. It is clear that more expected targets can be detected *via* LCB-YOLOv5s compared to the other methods. Additionally, Figures 14, 15 display the *mAP* (threshold is 0.5) of the six algorithms on the Visdrone2019 and DIOR-VAS datasets. The visual results of the original YOLOv5s and the LCB-YOLOv5s are demonstrated in Table 5. It can be intuitively seen that the proposed LCB-YOLOv5 algorithm has a better performance and higher robustness for

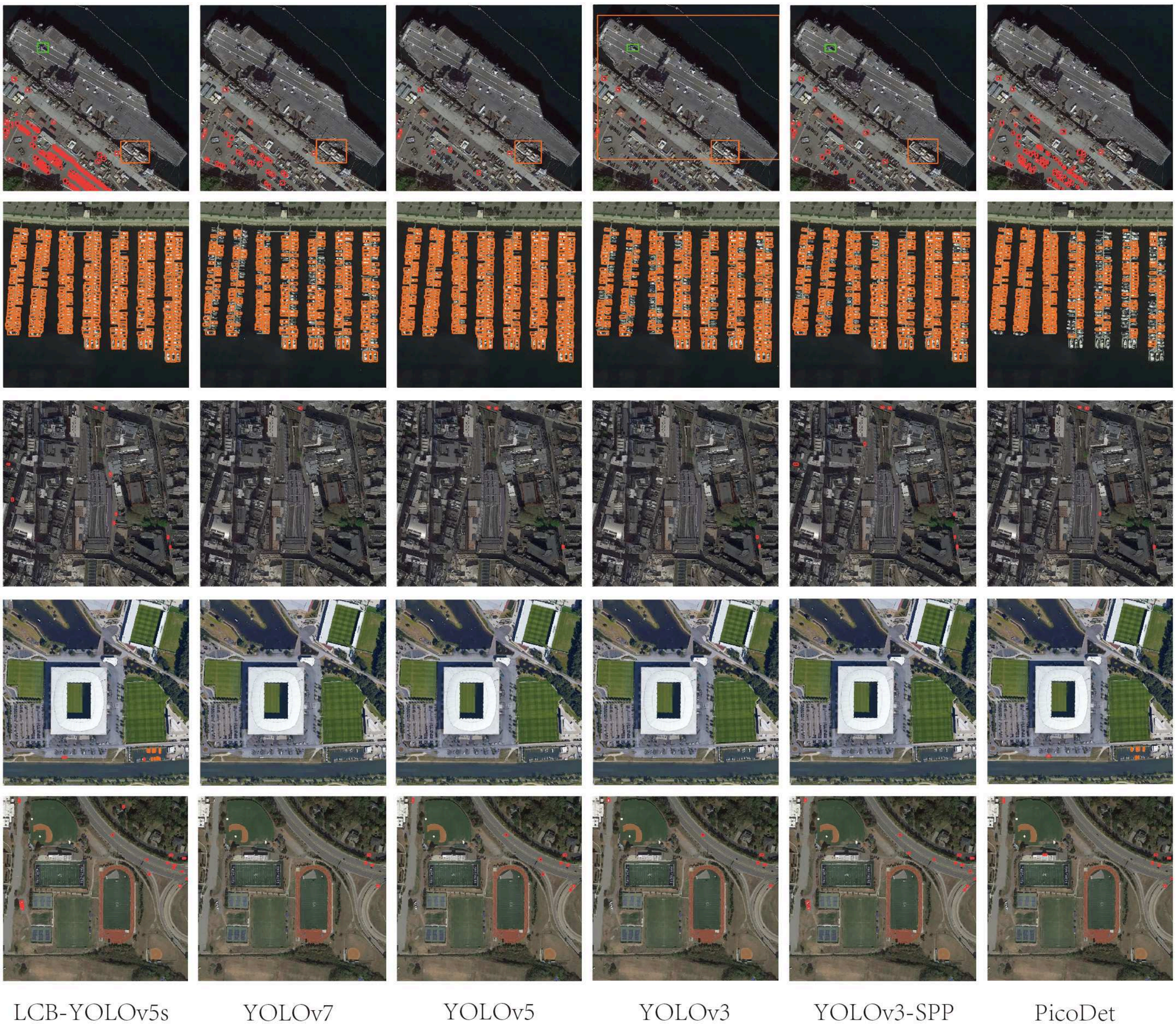


FIGURE 13
Comparison of the target detection of six different models on the DIOR-VAS dataset.

small target detection in remote sensing images. In particular, the LCB-YOLOv5s have a stronger ability in dense small target detection.

3.4. Results of ablation experiments

Ablation experiments are further performed to verify the optimization performance of each improved module. The EIoU loss function, LCB module, SPPS module, and Dres2 Module are introduced in the original network to construct the improved Model 1, improved Model 2, improved Model 3, and improved Model 4, respectively. In the improved model 5, the input size is 1,024, while all the mentioned modifications above are applied in the improved Model 6. The ablation results with the improved modules are listed in Table 6.

Compared with the original YOLOv5s network, the *mAP* of the model is improved by 1.3 percentage points in IM1, and the *mAP* of the models with IM3 and IM4 is increased by 0.9 and 0.6 percentage points, respectively. Moreover, the *mAP* of the model is improved by 11.8 percentage points with IM2. Meanwhile, when the input size is 1,024, the *mAP* of IM5 is also improved by 12.8 percentage points. Furthermore, when these six improvements are combined in IM6, the *mAP* is increased by 17.4 percentage points. The ablation experimental results strongly demonstrate that the proposed LCB-YOLOv5s model has a higher detection performance for small target detection with remote sensing images.

4. Conclusion

In this paper, an improved detection algorithm, called LCB-YOLOv5s, has been developed to detect small target objects in remote sensing images. The proposed algorithm comprises the LCB module *via* the combination of LSM and C3 modules, the SPPS module, and the Dres2 module in the feature extraction module to achieve multi-scale feature fusion. Furthermore, the input size of the network is increased and the output feature map size is set in various scales to improve the small target detection performance. Experiments have been performed on the DIOR and Visdrone2019 datasets to compare with other methods to verify the effectiveness of the proposed method for small target detection. Future work will continue to investigate small target detection and tracking under special and harsh circumstances with more general remote sensing datasets.

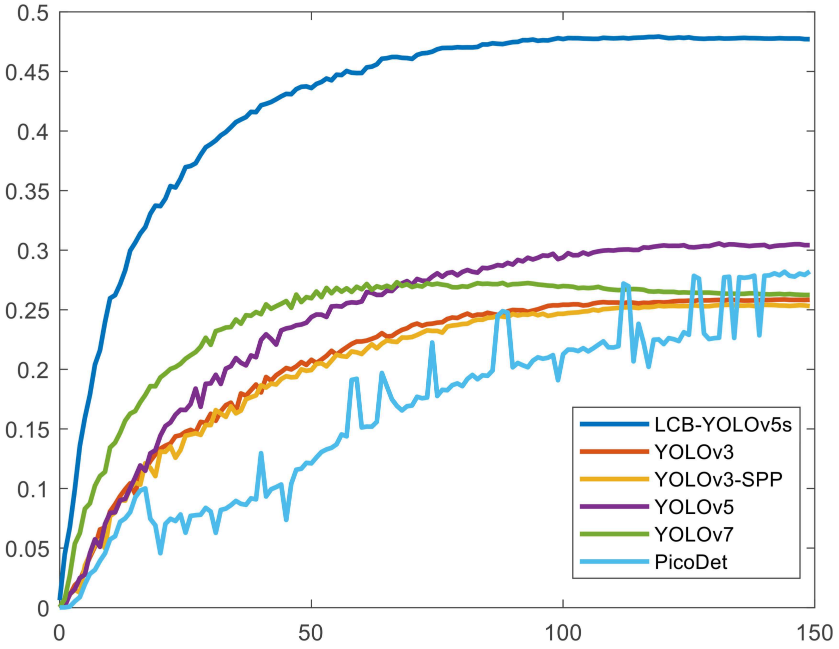


FIGURE 14
The *mAP* (threshold is 0.5) of the proposed method in comparison with the other five detection algorithms on the Visdrone2019 dataset.

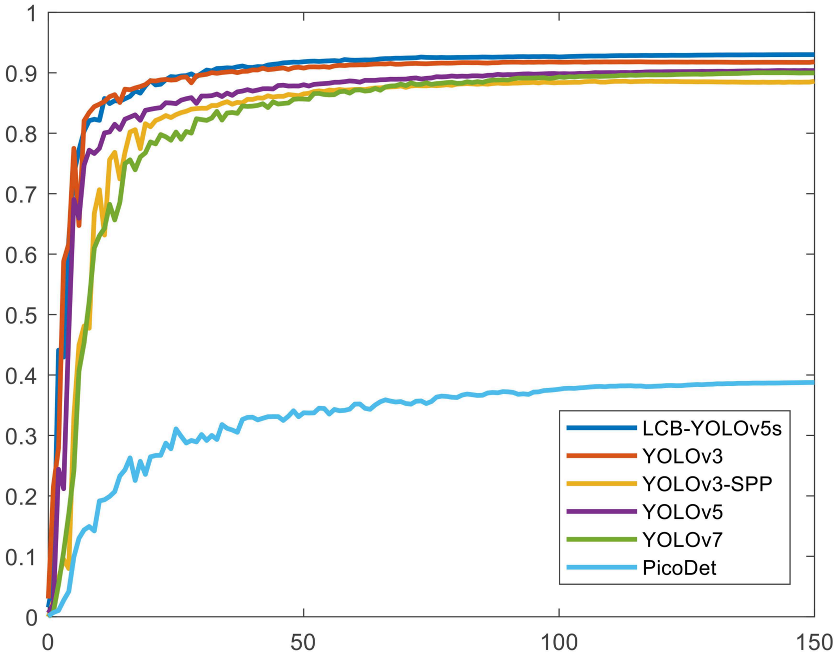


FIGURE 15
The *mAP* (the threshold is 0.5) of the proposed method in comparison with the other five detection algorithms on the DIOR-VAS dataset.

TABLE 5 Visual results of the small target detection on Visdrone2019 dataset.

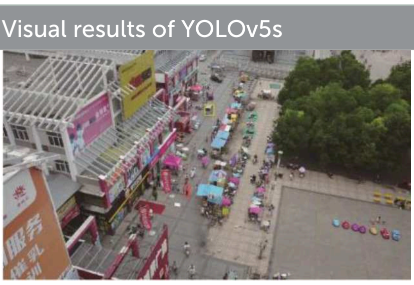
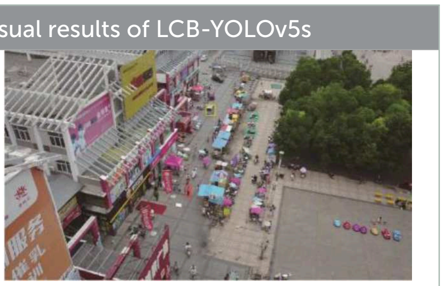
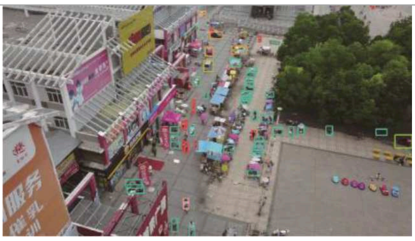
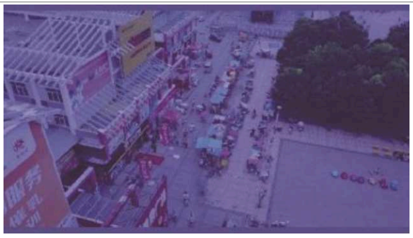
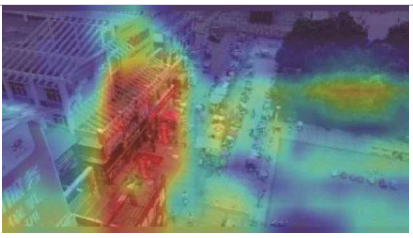
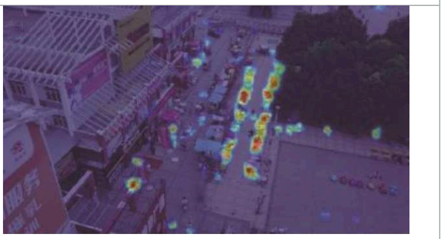
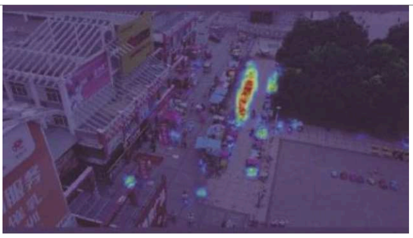
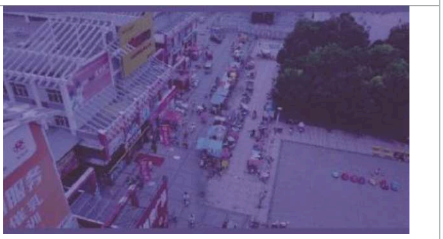
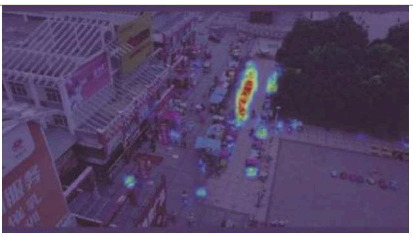
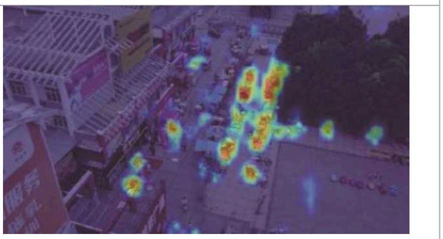
Categories	Visual results of YOLOv5s	Visual results of LCB-YOLOv5s
The original images		
Backbone		
Prediction head 1		
Prediction head 2		
Prediction head 3		

TABLE 6 Results of ablation experiments.

Model	EIOU	LCB	SPPS	Dres2	Input 1,024	*mAP*	Improvement (*mAP*)
LCB-YOLOv5s	×	×	×	×	×	30.5	-
Improved Model 1 (IM1)	√	×	×	×	×	31.8	+1.3
Improved Model 2 (IM2)	×	√	×	×	×	42.3	+11.8
Improved Model 3 (IM3)	×	×	√	×	×	31.4	+0.9
Improved Model 4 (IM4)	×	×	×	√	×	31.1	+0.6
Improved Model 5 (IM5)	×	×	×	×	√	43.3	+12.8
Improved Model 6 (IM6)	√	√	√	√	√	47.9	+17.4

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

Conceptualization and revising: WP, ZS, and KG. Methodology, experiments, and writing the original: WP and ZS. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported in part by the NSFC (62133001 and 61520106010) and the National Basic Research Program of China (973 Program: 2012CB821200 and 2012CB821201).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Chen, S.-B., Wei, Q.-S., Wang, W.-Z., Tang, J., Luo, B., Wang, Z.-Y., et al. (2021). Remote sensing scene classification *via* multi-branch local attention network. *IEEE Trans. Image Process.* 31, 99–109. doi: 10.1109/TIP.2021.3127851

Dong, X., Tian, J., and Tian, Q. (2022). A feature fusion airport detection method based on the whole scene multispectral remote sensing images. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 15, 1174–1187. doi: 10.1109/JSTARS.2021.3139926

Du, B., Sun, Y., Cai, S., Wu, C., and Du, Q. (2018). Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm. *IEEE Geosci. Remote Sens. Lett.* 15, 168–172. doi: 10.1109/LGRS.2017.2776899

Fan, J., Lee, J. H., Jung, I. S., and Lee, Y. K. (2021). "Improvement of object detection based on Faster R-CNN and YOLO," in *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications* (Jeju), 1–4.

Fu, J., Sun, X., Wang, Z., and Fu, K. (2020). An anchor-free method based on feature balancing and refinement network for multiscale ship detection in SAR images. *IEEE Trans. Geosci. Remote Sens.* 59, 1331–1344. doi: 10.1109/TGRS.2020.3005151

Guo, Y., Jia, X., and Paull, D. (2017). "A domain-transfer support vector machine for multi-temporal remote sensing imagery classification," in *2017 IEEE International Geoscience and Remote Sensing Symposium* (Fort Worth, TX: IEEE), 2215–2218.

Habibzadeh, M., Ameri, M., Sadat Haghighi, S. M., and Ziari, H. (2022). Application of artificial neural network approaches for predicting accident severity on rural roads (case study: tehran-qom and tehran-saveh rural roads). *Math. Probl. Eng.* 2022, 521470. doi: 10.1155/2022/5214703

Han, X., Zhong, Y., and Zhang, L. (2017). An efficient and robust integrated geospatial object detection framework for high spatial resolution remote sensing imagery. *Remote Sens.* 9, 666. doi: 10.3390/rs9070666

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE), 770–778.

Hou, B., Ren, Z., Zhao, W., Wu, Q., and Jiao, L. (2020). Object detection in high-resolution panchromatic images using deep models and spatial template matching. *IEEE Trans. Geosci. Remote Sens.* 58, 956–970. doi: 10.1109/TGRS.2019.2942103

Hu, Y., Li, X., Zhou, N., Yang, L., Peng, L., Xiao, S., et al. (2019). A sample update-based convolutional neural network framework for object detection in large-area remote sensing images, *IEEE Geosci. Remote Sens. Lett.* 16, 947–951. doi: 10.1109/LGRS.2018.2889247

Huang, L., Chen, C., Yun, J., Sun, Y., Tian, J., Hao, Z., et al. (2022). Multi-scale feature fusion convolutional neural network for indoor small target detection. *Front. Neurorobot.* 16, 881021. doi: 10.3389/fnbot.2022.881021

Huang, Z., Li, W., Xia, X., Wang, H., Jie, F., Tao, R., et al. (2022). LO-Det: lightweight oriented object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* 60, 5603515. doi: 10.1109/TGRS.2021.3067470

Jia, Y. (2000). Robust control with decoupling performance for steering and traction of 4WS vehicles under velocity-varying motion. *IEEE Trans. Control Syst. Technol.* 8, 554–569. doi: 10.1109/87.845885

Jia, Y. (2003). Alternative proofs for improved lmi representations for the analysis and the design of continuous-time systems with polytopic type uncertainty: a predictive approach. *IEEE Trans. Autom. Control.* 48, 1413–1416. doi: 10.1109/TAC.2003.815033

Jiang, W., Zhao, L., Wang, Y., Liu, W., and Liu, B.-D. (2021). U-Shaped attention connection network for remote-sensing image super-resolution. *IEEE Geosci. Remote Sens. Lett.* 19, 1–5. doi: 10.1109/LGRS.2021.3127988

Lawal, M. O. (2021). Tomato detection based on modified YOLOv3 framework. *Sci. Rep.* 11, 1–11. doi: 10.1038/s41598-021-81216-5

Li, H., and Man, Y. (2016). "Moving ship detection based on visual saliency for video satellite," in *2016 IEEE International Geoscience and Remote Sensing Symposium* (Beijing: IEEE), 1248–1250.

Li, K., Wan, G., Cheng, G., Meng, L., and Han, J. (2020). Object detection in optical remote sensing images: a survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* 159, 296–307. doi: 10.1016/j.isprsjprs.2019.11.023

Li, N., Cheng, L., Huang, L., Ji, C., Jing, M., Duan, Z., et al. (2021). Framework for unknown airport detection in broad areas supported by deep learning and geographic analysis. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 14, 6328–6338. doi: 10.1109/JSTARS.2021.3088911

Li, S., Xu, Y., Zhu, M., Ma, S., and Tang, H. (2019). Remote sensing airport detection based on end-to-end deep transferable convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* 16, 1640–1644. doi: 10.1109/LGRS.2019.2904076

Ling, J., Wang, H., Xu, M., Chen, H., Li, H., Peng, J., et al. (2022). Mathematical study of neural feedback roles in small target motion detection. *Front. Neurorobot.* 16, 984430. doi: 10.3389/fnbot.2022.984430

Liu, Y., Liao, Y., Lin, C., Jia, Y., Li, Z., Yang, X., et al. (2022). Object tracking in satellite videos based on correlation filter with multi-feature fusion and motion trajectory compensation. *Remote Sens.* 14, 2022. doi: 10.3390/rs14030777

Lu, X., Ji, J., Xing, Z., and Miao, X. (2021). Attention and feature fusion SSD for remote sensing object detection. *IEEE Trans. Instrum. Meas.* 70, 1–9. doi: 10.1109/TIM.2021.3118092

Lu, X., Zhang, Y., Yuan, Y., and Feng, Y. (2020). Gated and axis-concentrated localization network for remote sensing object detection. *IEEE Trans. Geosci. Remote Sens.* 58, 179–192. doi: 10.1109/TGRS.2019.2935177

Ma, H., and Wang, Y. (2022). Full information H2 control of borel-measurable Markov jump systems with multiplicative noises. *Mathematics* 10, 37. doi: 10.3390/math10010037

Meng, L. and Kerekes, J. P. (2012). Object tracking using high resolution satellite imagery. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 5, 146–152. doi: 10.1109/JSTARS.2011.2179639

Mikriukov, G., Ravanbakhsh, M., and Demir, B. (2022). "Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing* (Singapore: IEEE), 4463–4467.

Pei, W. (2022). Staring imaging attitude tracking control laws for video satellites based on image information by hyperbolic tangent fuzzy sliding mode control. *Comput. Intell. Neurosci.* 2022, 8289934. doi: 10.1155/2022/8289934

Pei, W., and Lu, X. (2022). Moving object tracking in satellite videos by kernelized correlation filter based on color-name features and Kalman prediction. *Wirel. Commun. Mob. Comput.* 2022, 9735887. doi: 10.1155/2022/9735887

Shakarami, A., Menhaj, M. B., Mahdavi-Hormat, A., and Tarrah, H. (2021). A fast and yet efficient YOLOv3 for blood cell detection. *Biomed. Signal Process. Control* 66, 102495. doi: 10.1016/j.bspc.2021.102495

Shi, P., Jiang, Q., Shi, C., Xi, J., Tao, G., Zhang, S., et al. (2021). Oil well detection *via* large-scale and high-resolution remote sensing images based on improved YOLO v4. *Remote Sens.* 13, 3243. doi: 10.3390/rs13163243

Tu, J., Gao, F., Sun, J., Hussain, A., and Zhou, H. (2021). Airport detection in sar images *via* salient line segment detector and edge-oriented region growing. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* 14, 314–326. doi: 10.1109/JSTARS.2020.3036052

Wang, J., Wang, Y., Wu, Y., Zhang, W., and Wang, Q. (2020). FRPNet: a feature-reflowing pyramid network for object detection of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* 99, 1–5. doi: 10.1109/LGRS.2020.3040308

Wang, M., Dong, Z., Cheng, Y., and Li, D. (2018). Optimal segmentation of high-resolution remote sensing image by combining superpixels with the minimum spanning tree. *IEEE Trans. Geosci. Remote Sens.* 56, 228–238. doi: 10.1109/TGRS.2017.2745507

Wang, P., Sun, X., Diao, W., and Fu, K. (2020). Fmssd: feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* 58, 3377–3390. doi: 10.1109/TGRS.2019.2954328

Wu, W., Yin, Y., Wang, X., and Xu, D. (2018). Face detection with different scales based on faster R-CNN. *IEEE T. Cybern.* 49, 4017–4028. doi: 10.1109/TCYB.2018.2859482

Xu, D. and Wu, Y. (2020). Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. *Sensors* 20, 4276. doi: 10.3390/s20154276

Yin, S., Li, H., and Teng, L. (2020). Airport detection based on improved Faster RCNN in large scale remote sensing images. *Sens. Imaging* 21, 1–13. doi: 10.1007/s11220-020-00314-2

Yu, Y., Yang, X., and Li, J. and Gao, X. (2022). A cascade rotated anchor-aided detector for ship detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* 60, 5600514. doi: 10.1109/TGRS.2020.3040273

Zhang, F., Du, B., Zhang, L., and Xu, M. (2016). Weakly supervised learning based on coupled convolutional neural networks for aircraft detection. *IEEE Trans. Geosci. Remote Sens.* 54, 5553–5563. doi: 10.1109/TGRS.2016.2569141

Zhang, G., Lu, S., and Zhang, W. (2019). Cad-net: a context-aware detection network for objects in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens*. 57, 10015–10024. doi: 10.1109/TGRS.2019.2930982

Zhang, P., Niu, X., Dou, Y., and Xia, F. (2017). Airport detection on optical satellite images using deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* 14, 1183–1187. doi: 10.1109/LGRS.2017.2673118

Zhang, Y., Ren, W., Zhang, Z., Jia, Z., Wang, L., Tan, T., et al. (2021). Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* 506, 146–157. doi: 10.1016/j.neucom.2022.07.042

Zhong, Y., and Zheng, Z. Ma, A., Lu, X., and Zhang, L. (2020). Color: cycling, offline learning, and online representation framework for airport and airplane detection using gf-2 satellite images. *IEEE Trans. Geosci. Remote Sens*. 58, 8438–8449. doi: 10.1109/TGRS.2020.2987907

Zhou, L., Deng, G., Li, W., Mi, J., and Lei, B. (2021). A lightweight SE-YOLOv3 network for multi-scale object detection in remote sensing imagery. *Int. J. Pattern Recognit. Artif. Intell.* 35, 2150037. doi: 10.1142/S0218001421500373

Zhu, P., Du, D., Wen, L., Bian, X., Ling, H., Hu, Q., et al. (2019). "Visdrone-vid2019: The vision meets drone object detection in video challenge results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (Seoul: IEEE), 227–235.

Zhu, X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., et al. (2017). Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* 5, 8–36. doi: 10.1109/MGRS.2017.2762307

TYPE Original Research
PUBLISHED 15 February 2023
DOI 10.3389/fnbot.2023.1093718

OPEN ACCESS

EDITED BY
Chen Qiao,
Xi'an Jiaotong University, China

REVIEWED BY
Lourdes De Martínez,
Panamerican University Benito Juárez, Mexico
Miao Li,
Swiss Federal Institute of Technology Lausanne, Switzerland
Peng Wang,
Institute of Automation (CAS), China
Gongfa Li,
Wuhan University of Science and Technology, China

*CORRESPONDENCE
Yanhe Zhu
✉ yhzhu@hit.edu.cn

RECEIVED 09 November 2022
ACCEPTED 30 January 2023
PUBLISHED 15 February 2023

CITATION
Zhao S, Zheng T, Sui D, Zhao J and Zhu Y (2023) Reinforcement learning based variable damping control of wearable robotic limbs for maintaining astronaut pose during extravehicular activity.
Front. Neurorobot. 17:1093718.
doi: 10.3389/fnbot.2023.1093718

Reinforcement learning based variable damping control of wearable robotic limbs for maintaining astronaut pose during extravehicular activity

Sikai Zhao, Tianjiao Zheng, Dongbao Sui, Jie Zhao and Yanhe Zhu*

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, China

As astronauts perform on-orbit servicing of extravehicular activity (EVA) without the help of the space station's robotic arms, it will be rather difficult and labor-consuming to maintain the appropriate position in case of impact. In order to solve this problem, we propose the development of a wearable robotic limb system for astronaut assistance and a variable damping control method for maintaining the astronaut's position. The requirements of the astronaut's impact-resisting ability during EVA were analyzed, including the capabilities of deviation resistance, fast return, oscillation resistance, and accurate return. To meet these needs, the system of the astronaut with robotic limbs was modeled and simplified. In combination with this simplified model and a reinforcement learning algorithm, a variable damping controller for the end of the robotic limb was obtained, which can regulate the dynamic performance of the robot end to resist oscillation after impact. A weightless simulation environment for the astronaut with robotic limbs was constructed. The simulation results demonstrate that the proposed method can meet the recommended requirements for maintaining an astronaut's position during EVA. No matter how the damping coefficient was set, the fixed damping control method failed to meet all four requirements at the same time. In comparison to the fixed damping control method, the variable damping controller proposed in this paper fully satisfied all the impact-resisting requirements by itself. It could prevent excessive deviation from the original position and was able to achieve a fast return to the starting point. The maximum deviation displacement was reduced by 39.3% and the recovery time was cut by 17.7%. Besides, it also had the ability to prevent reciprocating oscillation and return to the original position accurately.

KEYWORDS

extravehicular activity, reinforcement learning, wearable robotic limbs, variable damping control, modular robot

1. Introduction

With the progress of robot technology and artificial intelligence, space exploration has ushered in rapid development (Chien and Wagstaff, 2017; Jacobstein et al., 2017; Lester et al., 2017). The world's space powers began to carry out manned space activities around the International Space Station (ISS) (Flores-Abad et al., 2014; Jiang et al., 2017; Ruttley et al., 2017). In addition, commercial space tourism has gradually become a new highlight of manned space development in recent years. Some private manned space companies have successfully completed several commercial space trips (Webber, 2013; Chang, 2015). It can be established that manned space engineering will play an increasingly important role in space exploration and on-orbit servicing. Therefore, there will be higher requirements for astronauts' operations in the outer space environment.

Extravehicular activity (EVA) refers to when astronauts wear spacesuits to perform tasks outside the spacecraft, which is the key technology of manned space engineering. The environment of EVA is extreme, work intensity is high, and the operation process is complex. These issues greatly restrict the astronaut's EVA time and success rate. In recent years, various robot-intelligent technologies have been applied to assist in the reduction of work intensity and improve workability (Zhang et al., 2020a; Wang S. et al., 2022). However, they have not been popularized and applied in the space field. For on-orbit servicing of EVA, there are two main modes of extravehicular movement. One is that astronauts move in vast space through the robotic arm of the space station (Nokleby, 2007; McHenry et al., 2020). The other is that astronauts climb by themselves with the help of safety ropes. In the former scenario, astronauts' lower limbs are fixed to the robotic arm, which can provide a foot restrictor and liberate the upper limbs to accomplish tasks. However, there are some areas where the space station's robotic manipulator cannot reach. In these areas, with the lack of a space robotic manipulator, astronauts will have to move and work by themselves under the protection of safety ropes. When working in this situation, it is significantly hard for them to maintain a suitable position when they suffer some form of impact. In order to maintain stability, they need to exert force with one or two hands, which not only increases energy consumption but also greatly limits work efficiency. Thus, astronauts need additional devices that assist them in resisting impact and maintaining position during the process of EVA. There is much research on trajectory planning and control of robotic manipulators at home and abroad (Zhang et al., 2020b, 2022; Zhao et al., 2022). However, these large dedicated robots and equipment have high launch costs and low utilization rates. Several astronaut-assisting robots have been developed, including humanoid robots (Diftler et al., 2011; Ackerman, 2019), on-orbit modeling robots (Zykov et al., 2007; Post et al., 2021), and wearable-assisting robots (Hall, 2013; Zhao et al., 2021). The primary application purpose of these devices is to provide astronauts with operation or strength enhancement assistance for on-orbit servicing. In addition, they are either still in the conceptual design stage or can only be used inside the cabins of the ISS. Thus, none of them can provide astronauts with the ability to withstand external impact. Although some impact-resisting methods of the space station's robotic arms have been studied (Su et al., 2020; Liu et al., 2021; Olivieri et al., 2021; Raina et al., 2021; Wang X. et al., 2022), they are only used for the robotic arms themselves or on missions to capture free-flying objects, and not in helping astronauts. In addition, the main problem is that large space manipulators cannot be applied to all task scenes.

Wearable robotic limbs can provide a new method for assisting astronauts in performing tasks, especially those carrying out extravehicular work alone. The robotic limb can act as an extra limb of the astronaut and improve the wearer's abilities of perception and operation. In this way, it has the potential to reduce the astronaut's physical exertion and consumption in extravehicular activities and improve the success rate of on-orbit servicing tasks. Considering the safe and comfortable operation requirements of astronauts, the wearable robotic limb system is expected to have the following impact resistance capabilities: (1) Deviation resistance: The deviation after impact cannot be too large. It is very dangerous to deviate too far from the operating position. In case of an emergency, astronauts should have the ability to grasp the handrail; (2) Fast return: After reaching the maximum deviation position, it can quickly return to the initial position. It is helpful to extend the effective working time of EVA; (3) Oscillation resistance: After the system quickly restores to the initial position, it is necessary to prevent reciprocating oscillation relative to the initial position, which will cause system instability and physical discomfort; and (4) Accurate return: The system must be able to return to the original position after impact. Otherwise, astronauts need to make additional manual adjustments, which indirectly increases the difficulty and physical exertion of the task.

As far as we know, no similar concepts of robots for assisting astronauts have been proposed yet. The purpose of this paper is to propose a variable damping control method based on a reinforcement learning algorithm for wearable robotic limbs, in which the virtual damping is trained to be adjustable to meet the impact resistance requirements proposed above. The method was verified in a simulation environment, which ensured that the robotic limb system has the ideal impact-resisting ability. The rest of the paper is organized as follows: Section 2 introduces the basic composition of the wearable robotic limbs for astronauts and explains the variable damping control method based on Reinforcement Learning; Section 3 presents the simulation results and evaluation; and Section 4 summarizes the whole work, analyzes the application limitations, and outlines plans for future work.

2. Materials and methods

2.1. Wearable robotic limb system

Astronauts can work in orbit outside the cabin of the ISS in two main ways. The first is that the astronaut's feet are attached to the end of the space station's robotic arm. As shown in **Figure 1A**, the space station's robotic arm provides the astronaut with a foot restrictor, so that the astronaut can maintain the desired position through the lower limbs. Meanwhile, the upper limbs and hands are free to perform tasks. The second is that the astronaut is connected to the working area via a safety rope without using the space station's robotic arm. In this case, there is no reliable anchor point such as the foot restrictor. If the astronaut wants to maintain a proper working position, one hand is needed to maintain that position, as shown in **Figure 1B**. In this situation, it is not suitable for the astronaut to operate with both hands simultaneously, and the astronaut cannot perform complex operational tasks that call for two-handed cooperation. In addition, it will consume considerable energy and reduce the EVA time.

In view of the above shortcomings, we proposed a wearable robotic limb system that can be fixed onto the astronaut's backpack as additional arms to assist in moving and operating outside the ISS. The system is named AstroLimbs (Zhao et al., 2021). **Figures 1C,D** show the rendered views of the front and back sides of the AstroLimbs, respectively. The wearing display of the robotic limb system is shown in **Figure 1E**. Based on the modular design concept, each robotic limb is composed of six identical basic modules connected in series. The modular design concept is suitable for space engineering, with more

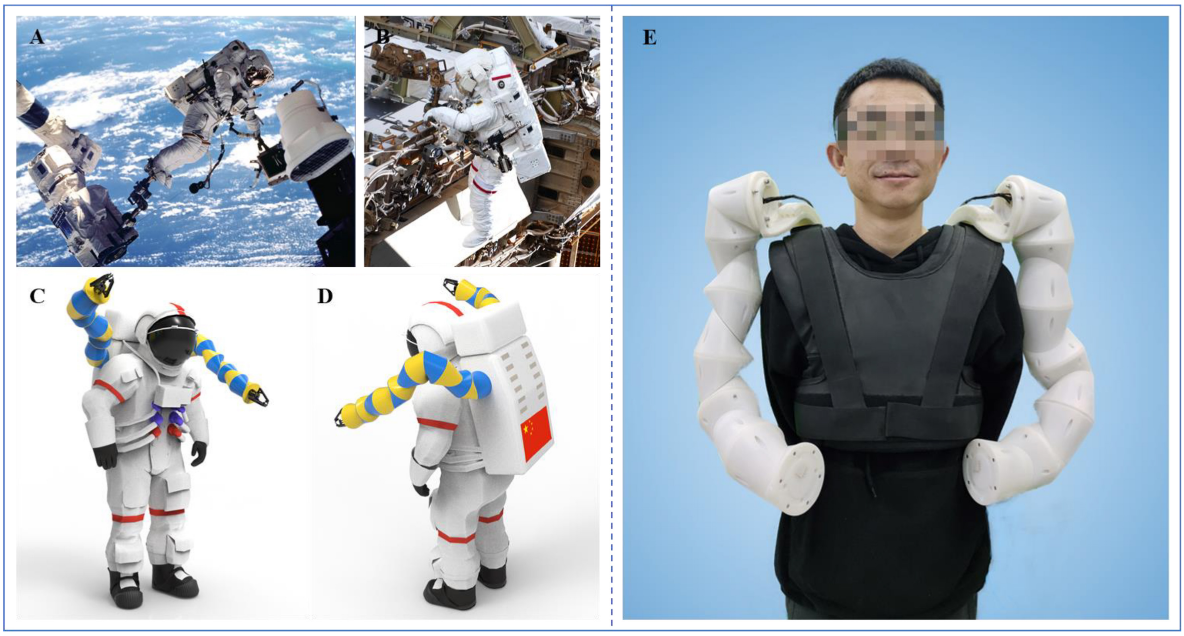


FIGURE 1
Wearable robotic limbs for astronauts. **(A)** Performing EVA with the help of the space station's robotic arm (Mohon, 2014). **(B)** Performing EVA without the space station's robotic arm (Garcia, 2019). **(C)** The rendered view of the front side of the wearable robotic limbs for astronauts. **(D)** The rendered view of the back side of the wearable robotic limbs for astronauts. **(E)** Wearing display of the robotic limbs for astronauts.

convenient assembly, better interchangeability, and improved fault tolerance. The end faces of both submodules are equipped with the connection mechanism. Two basic modular units can be connected in series via the connection mechanism. Each basic module serves as a joint of the robotic limb. This means that each robotic limb has six degrees of freedom. The AstroLimbs can be worn on the astronaut's backpack, moving and working with the wearer. It acts as a working partner for the wearer during EVA, just like another astronaut. As the outer space environment is almost weightless, the weight and mass of the robotic system will not be applied to the astronaut.

2.2. Variable damping control

2.2.1. Model building

In order to achieve the robotic limb's ability to maintain the astronaut's posture during EVA, the variable damping control method based on the Q-learning algorithm was proposed. Prior to the reinforcement learning training, it was necessary to model and simplify the astronaut system with the robotic limbs, which could function faster in the simulation environment, as shown in Figure 2. While the astronaut works outside the ISS cabin, one robotic limb holds the handrail to maintain the position in the working area. Under this condition, the handrail was considered as a fixed end and the end of the robotic limb was simplified to connect to that fixed end. The astronaut and the other robotic limb were combined and simplified into an end-load system, where the second robotic limb mainly provides auxiliary functions, such as tool delivery and operational support. As shown in Figure 2, they were reduced to a green solid ball at the end of the robotic limb. The blue ellipses represent the links of the robotic limb, and these links are connected by rotating joints, which are represented by the solid blue points. Each robotic limb had six degrees of spatial freedom. The fixed end was equal to the handrail of the ISS. The Cartesian coordinate system, which is the absolute coordinate system, was attached to the fixed end. Combined with the forward kinematics of the robotic limb, the end-load movement information for Cartesian space could be obtained in real time.

In addition, this model could also be split into two systems. One was the load system and the other was the robotic limb system without the load. Based on the model, the variable virtual restoring force was introduced to control the load for impact resistance and maintenance of position. In combination with the Q-learning algorithm, the variable damping controller was formed. The virtual restoring force was taken as an external force of the robotic limb. Finally, based on its dynamics, the virtual restoring force could be transformed into the control torque of each joint. In this way, the robotic limb could realize its position-maintaining control to help the astronaut.

2.2.2. Variable damping control for end load

In order to achieve the optimum motion characteristics of the robotic limb end after impact, the most straightforward method was to determine the conversion relationship between the motion characteristics of the joint space and the end Cartesian space. It was necessary to discover the configuration changes of the limb in real time and calculate the equivalent moment of each joint inertia. The calculated quantity of the overall process was too high. Thus, the variable damping control method based on the virtual restoring force was introduced. For the load system, it was possible to obtain its absolute movement information in relation to the Cartesian space in real time. In this case, the load could be considered as an unconstrained spatial load that was only controlled by the virtual restoring force, so as to meet the proposed requirements for impact resisting. As shown in Figure 3, the virtual restoring force acted on

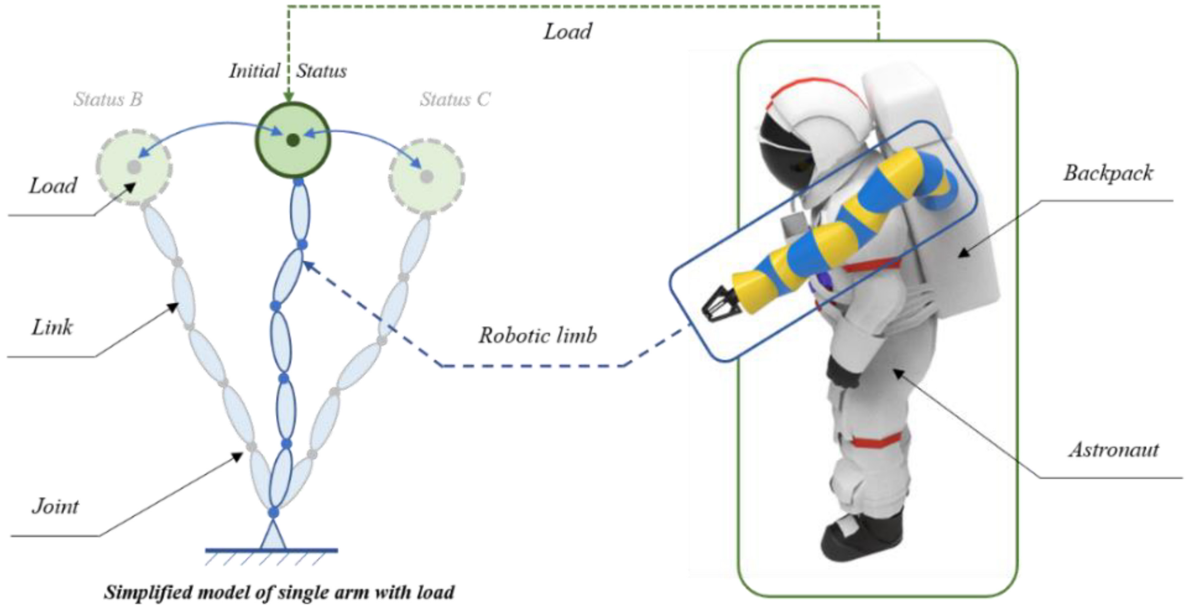


FIGURE 2
Simplified model of the astronaut and the robotic limb system.

the mass center of the load, so that the load tended to move back to its original position. Its value varied in real time, which was related to the motion state of the load (p_t, v_t). The mapping function f_{RL} between the virtual restoring force and the movement status could be achieved by the Q-learning algorithm.

For the load in weightlessness, in order to reduce the deviation and bring it back to the original position, a virtual restoring force based on the spring damping model was proposed. Its virtual damping coefficient could change adaptively, as shown in Figure 3. The change between the real-time state of the load and the initial state was used as the input of the virtual restoring force, and the virtual restoring force was mainly composed of the virtual spring tension and damping force, which can be shown as follows:

$$F_r = K \cdot X(t) + D(t) \cdot \dot{X}(t) \tag{1}$$

where F_r represents the virtual restoring force, K is the virtual spring stiffness coefficient, $D(t)$ is the virtual damping coefficient, $X(t)$ is the displacement relative to the initial position after impact, and $\dot{X}(t)$ is the velocity after impact. When the spatial load was impacted in any direction, the corresponding state changes occurred in the three-dimensional space, such as in *Status B* or *C* as shown in Figure 3. The spring damping system was applicable. That is to say, the virtual restoring force generated was always in a straight line with the displacement of the load in relation to the initial state.

For the introduced spring damping system, the corresponding impedance characteristics could be obtained by adjusting the appropriate stiffness coefficient K and damping coefficient D(t) according to the desired system characteristics. However, the fixed stiffness and damping coefficient could not simultaneously satisfy the overall impact resistance requirements. When the stiffness was fixed, if the damping coefficient was too small, the load-displacement was too large. If the damping coefficient was too large, the recovery speed after impact was too slow. Therefore, the damping coefficient was particularly critical for maximal deviation and recovery time. Considering the practical application of wearable robotic limbs, it was used to hold the handrail of the cabin to stabilize the position of the astronaut when working in a fixed spot. In this case, it was hoped that the equivalent system had a relatively large stiffness. At this time, if the method of variable stiffness was adopted, the stiffness of the system could be reduced, which was not conducive to the astronaut maintaining position. Therefore, the variable damping control method was selected in this paper. For the problem that the virtual restoring force of the fixed damping method could not fully meet the impact resistance requirements, the variable damping controller could change the virtual damping value appropriately depending on the real-time movement state, so as to meet the impact resisting requirements in different states.

2.2.3. Reinforcement learning

When it comes to tackling serialized decision-making issues in unknown contexts, reinforcement learning offers clear advantages. Q-learning is one of the reinforcement learning algorithms and can be used to adaptively learn the virtual damping of load movement in a weightless environment. Therefore, the state of load was divided based on the designed working environment, and the fundamental action was planned. Moreover, the reward function in the task-learning process was proposed.

Reinforcement learning is an overall process that refers to the agent's trial, evaluation, and action memory (Clifton and Laber, 2020; Chen et al., 2022; Cong et al., 2022; Li et al., 2022). The agent's learning maps from environment state to action, causing it to reap the greatest rewards after carrying out a particular action. This learning process will make the agent perform best under some preset evaluation rules. The Q-learning algorithm is one of the evaluation rules for the agent to choose a specific action in the present state, which is an action-utility function. Q is short for the word quality, which serves as high-quality feedback for each action and provides the agent with action memory (Ohnishi et al., 2019; Hutabarat et al., 2020). The Q-learning algorithm is excellent for model-free

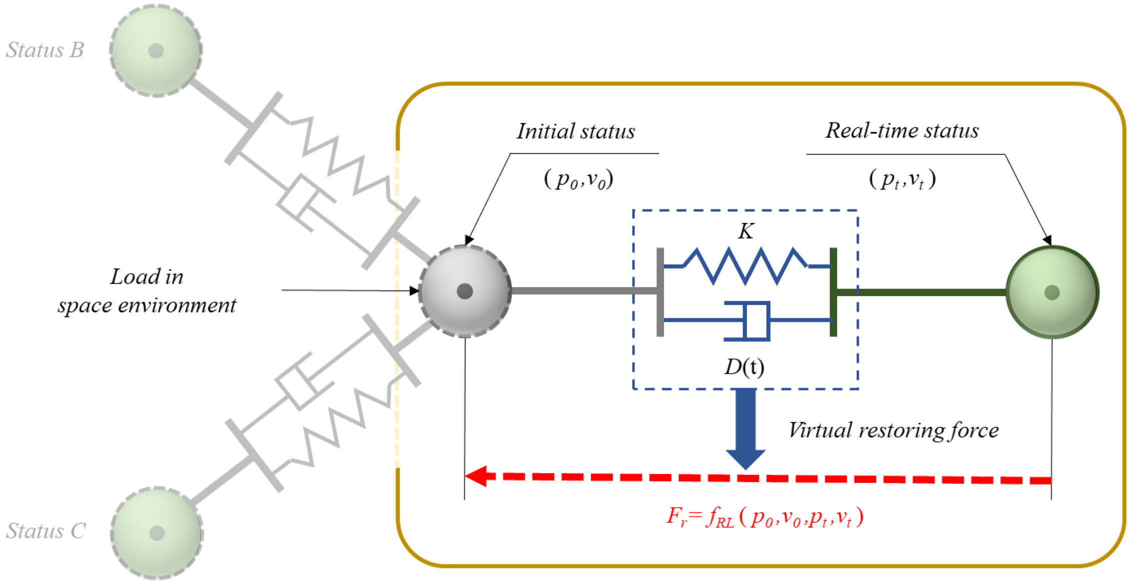


FIGURE 3
Variable damping control principle for load in weightless space.

autonomous motion planning when the number of states and actions in the learning process is limited (Clifton and Laber, 2020).

The following equation describes the agent's corresponding evaluation value after performing the action each time in a particular state:

$$Val = max_a Q(s, a) \tag{2}$$

Where *s* denotes the current state, *a* is the action that can be taken in the current state, and *Val* is the evaluated maximum value corresponding to this action under the circumstances of the current state s and action a. In light of this value, the agent can determine the action to execute in this step.

The core of the Q-learning algorithm is the process of constantly updating the evaluation value *Val* in Equation 2 based on continuous trial training:

$$Q'(s,a) \Leftarrow Q(s,a) + \lambda \left[R(s,a) + \eta \cdot max_{a'} Q(s',a') - Q(s,a) \right] \tag{3}$$

where *R* represents the reward value that can be obtained by executing action *a* in the current state *s*, *s'* is the new state of the agent after executing action *a*, *a'* is the possible action in state *s'*, λ is the learning efficiency (λ = 0.01), and η serves as the discount factor (η = 0.9).

First, the training was conducted in a single dimension, which simplified the load movement process. Based on the position and velocity information in relation to the Cartesian space, the motion state of the load determined the state of the Q-learning. The following equation provides the definition of the state value:

$$State = f\left(P, Flag\_v\right) \tag{4}$$

where *State* represents the load motion state, *P* is the displacement compared to its initial position, *Flag_v* denotes the velocity direction identification value depending on both the displacement and velocity direction, which can be expressed as the following Equation 5:

$$Flag\_v = \begin{cases} 1 & \vec{P} \cdot \vec{V} \geq 0 \\ 0 & \vec{P} \cdot \vec{V} < 0 \end{cases} \tag{5}$$

where $\vec{P}$ is the real-time displacement vector, $\vec{V}$ is the real-time speed vector.

In order to improve the efficiency of reinforcement learning, the displacement range was discretized. To guarantee applicability, displacement values outside the valid range were incorporated into adjacent state intervals. And the corresponding relationship between the acquired state and the movement state of the load is shown in Equation 6:

$$State = \left(\left\lceil {}^{P}/_{d} \right\rceil + s_{int} \right) + n \cdot Flag\_v \tag{6}$$

where *d* is the interval step size for displacement range, s_{int} is the state offset value designed to count state values from zero, and *n* represents the total number of states regardless of velocity direction. $\left\lceil {}^{P}/_{d} \right\rceil$ stands for the result of rounding up the ratio of *P* to *d*, which is the smallest integer greater than the ratio.

As the load had no gravity in a weightless environment, the control model could be equivalent to a spring-damping model. The force generated by the virtual spring and damping directly acted on the mass center of the load, so the virtual force generated by the real-time virtual spring tension and damping force after impact could be obtained. Thus, the load system's stiffness-damping characteristics were simulated to achieve optimal motion control. According to the simplified model, the virtual stiffness was designed to be a fixed value, so that the virtual restoring force of the load was proportional to its displacement value.

To avoid excessive displacement, oscillation, and failure to return to its original position after impact, it was necessary to change the virtual damping according to different states. Using the same discrete design idea as the motion state, the maximum virtual damping value was designed to be 600 and the interval step size was 150. Thus, the damping value could be used as an optional action in the Q-learning process in five cases, as shown in Equation 7:

$$Action=\{0,150,300,450,600\} \tag{7}$$

During the training process, the agent received a reward for each episode in which they interacted with the environment. For the process that the load suffered an impact in weightless space, it deviated from the original position. Under the action of the virtue spring tension and damping force, it could return to the original position after reaching the maximum deviation. According to the desired impact resistance requirements, the farther the load deviated from the initial position, the weaker its ability was to prevent oscillation, and fewer rewards were given. If the load got closer to the initial position, it obtained more rewards. Therefore, it made sense to take the negative value of the deviation distance as the reward, which can be expressed by the following equation:

$$R=-dis=-\sqrt{x_t^2+y_t^2+z_t^2} \tag{8}$$

where R is the reward value received for a particular action, *dis* is the distance value in relation to the starting position, and x_t, y_t, *and* z_t are the components of the real-time position.

In the training process, when the robot was in the initial position, the reward R obtained by the agent was zero. As the reward value R was designed to be non-positive, it meant that the reward value of the agent was the maximum in the initial state. After suffering an impact, the robot generated a position deviation. The reward value decreased as the deviation increased. It meant that the agent was punished.

2.2.4. Application on robotic limbs

The impact force applied on an astronaut outside the space station is three-dimensional and can come from any direction. As a result, the displacement and velocity directions of the load do not lie in a uniform line with respect to its initial state. Then the one-dimensional variable damping control method based on the Q-learning algorithm could not completely solve the issue. As the displacement and velocity directions were not in a straight line, a velocity vector was generated in the direction perpendicular to the displacement vector. The system eventually reached equilibrium as a result of the virtual restoring force acting in the displacement direction. The load then moved uniformly around the initial position, and the virtual restoring force provided the load with centripetal acceleration. However, the load was not able to return to the initial position.

In order to solve the above issue, a speed-decoupling control method based on one-dimensional control was purposed. The improved control principle is shown in Figure 4. The overall concept of this method was to carry out adaptive control in different dimensions through orthogonal decoupling of velocity. In Figure 4, the gray sphere represents the initial state of the load, and the green sphere represents the real-time movement status after impact. The line between the two states is the displacement direction, which was recorded as the Y direction. The positive Y direction pointed to the direction away from the initial position. The direction perpendicular to the Y direction was marked as the X direction. The selection of positive X direction is shown in Figure 4, which made no difference to the outcome. Since the motion state of the load after impact changed in real time, the X and Y directions also varied continuously. However, the Y direction could be uniquely determined depending on the displacement direction. When the Y direction was fixed, the X direction then became uniquely determined. The two directions could be determined at any time, even though they were constantly varying in real time. These two real-time directions were the base for orthogonal decoupling velocity. It can be seen from Figure 4 that the load speed V was orthogonally decoupled along the X and Y directions to obtain the velocity component V_x and V_y, respectively. The Y direction was the key direction for the load to return to the original position after suffering an impact. It was hoped that the load could resist impact in this direction. Therefore, the variable damping controller based on the reinforcement learning method was adopted in the Y direction. When V_x became zero, the issue normally transformed into the fundamental problem of impact resisting control for a single direction. Therefore, the control method in this direction was relatively simple, that is to set a large fixed damping coefficient. The velocity in this direction could be quickly reduced to zero as soon as possible.

For the unconstrained load model, the real-time restoring force was virtual and this hypothetical force in the simulation environment had no actual force application object. The load model and the robotic limb model were combined using this virtual force as a bridge. In order to ensure the robotic limb end had the same impact resistance performance as the load model, the force application object of the virtual restoring force should be the robotic limb itself. Hence, the problem was changed into the end force control issue of the series manipulator with six degrees of freedom. In combination with the dynamics of the robotic limb, the joint control torque for the real-time virtual restoring force could be obtained, so as to realize the impact resistance ability of the end load.

The magnitude of the virtual restoring force used to control the end load was in relation to the real-time motion state of the end load, as shown in Equation 9:

$$F_r=f_{RL}\left(p_0,v_0,p_t,v_t\right) \tag{9}$$

where F_r is the virtual restoring force acting on the rigid end of the robotic limb, p_0 is the initial displacement, v_0 is the initial velocity, p_t is the real-time displacement after impact, and v_t is the real-time velocity after impact.

Finally, the virtual restoring force of the robotic limb end was brought into the dynamic equation, so that the joint space control torque could be obtained and the impact-resisting control of the robotic limb end could be realized.

A framework of the variable damping control method to further explain the control method is shown in Figure 5. The combined system of an astronaut with a robot was modeled and simplified. With the help of system dynamics and coordinate transformation, the controller enabled the robotic limb end to resist impact. According to Figure 5, F_t stands for the impact applied on the system. F_r is the virtual restoring force originating from the variable damping controller based on the reinforcement learning method. τ is a six-dimensional vector, which stands for the torque of each joint. $\theta, \dot{\theta}, \ddot{\theta}$ is the motion information of each joint. p_t and v_t are the displacement and velocity of the robotic limb end, respectively.

Considering the practical application of wearable robotic limbs, they were used to hold the handrail of the cabins to stabilize the position of the astronaut when working in a fixed spot. In this case, it was hoped that the equivalent system of the robotic limbs and

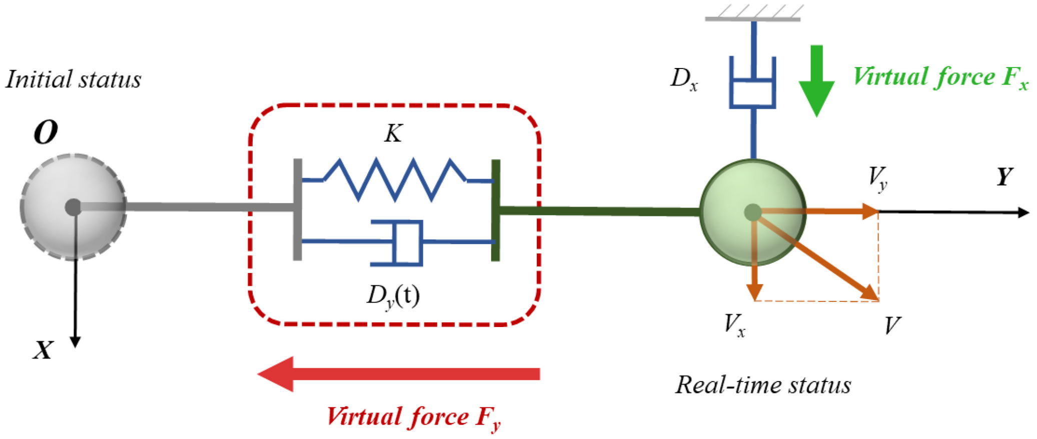


FIGURE 4
Schematic diagram of three-dimensional impact resistance of the load.

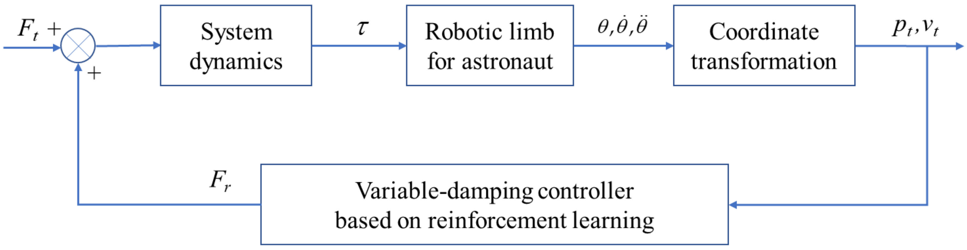

F_t +
+
System dynamics
τ
Robotic limb for astronaut
$\theta,\dot{\theta},\ddot{\theta}$
Coordinate transformation
p_t, v_t
F_r
Variable-damping controller based on reinforcement learning

FIGURE 5
Framework of the variable-damping control method.

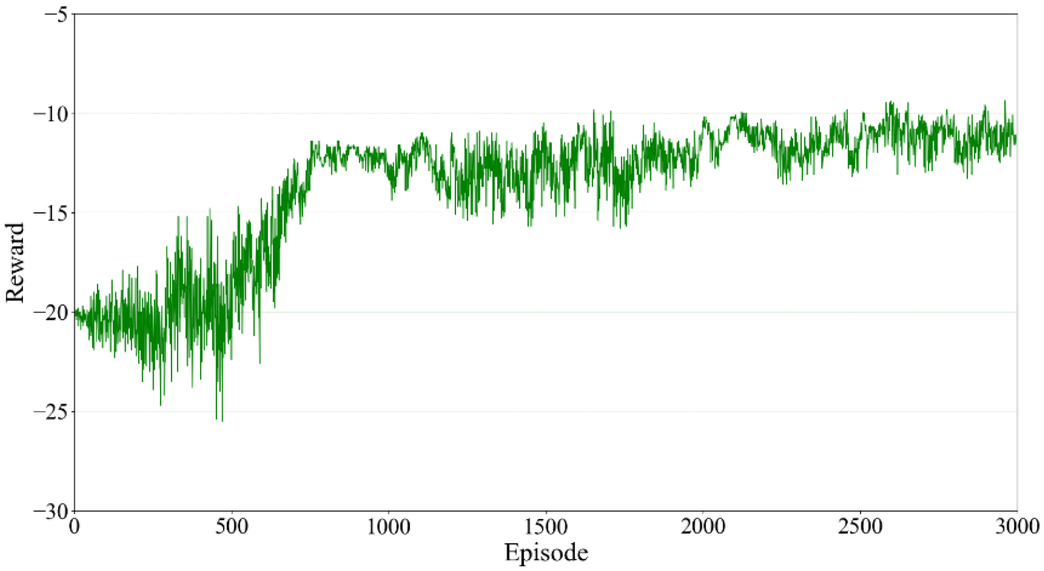


FIGURE 6
Reward value of each episode for the agent.

the astronaut had a relatively significant stiffness. At this time, if the method of variable stiffness was adopted, the stiffness of the system would be reduced, which would not be conducive to the astronaut maintaining position. Therefore, the variable damping control method was selected in this paper.

The core of the variable damping controller was that there was always a damping term in the system, and the damping coefficient could be appropriately changed according to the motion effect produced by the external impact. In addition, as the system deviated from its original position, the damping coefficient increased to

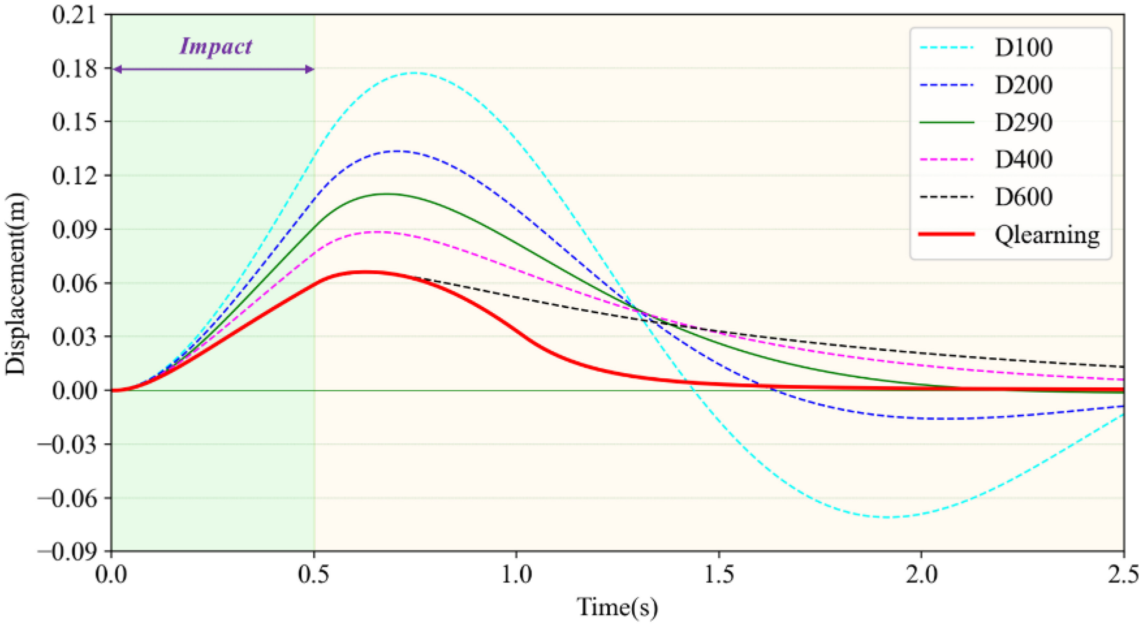


FIGURE 7
Comparison of load recovery trajectories under different damping values after impact.

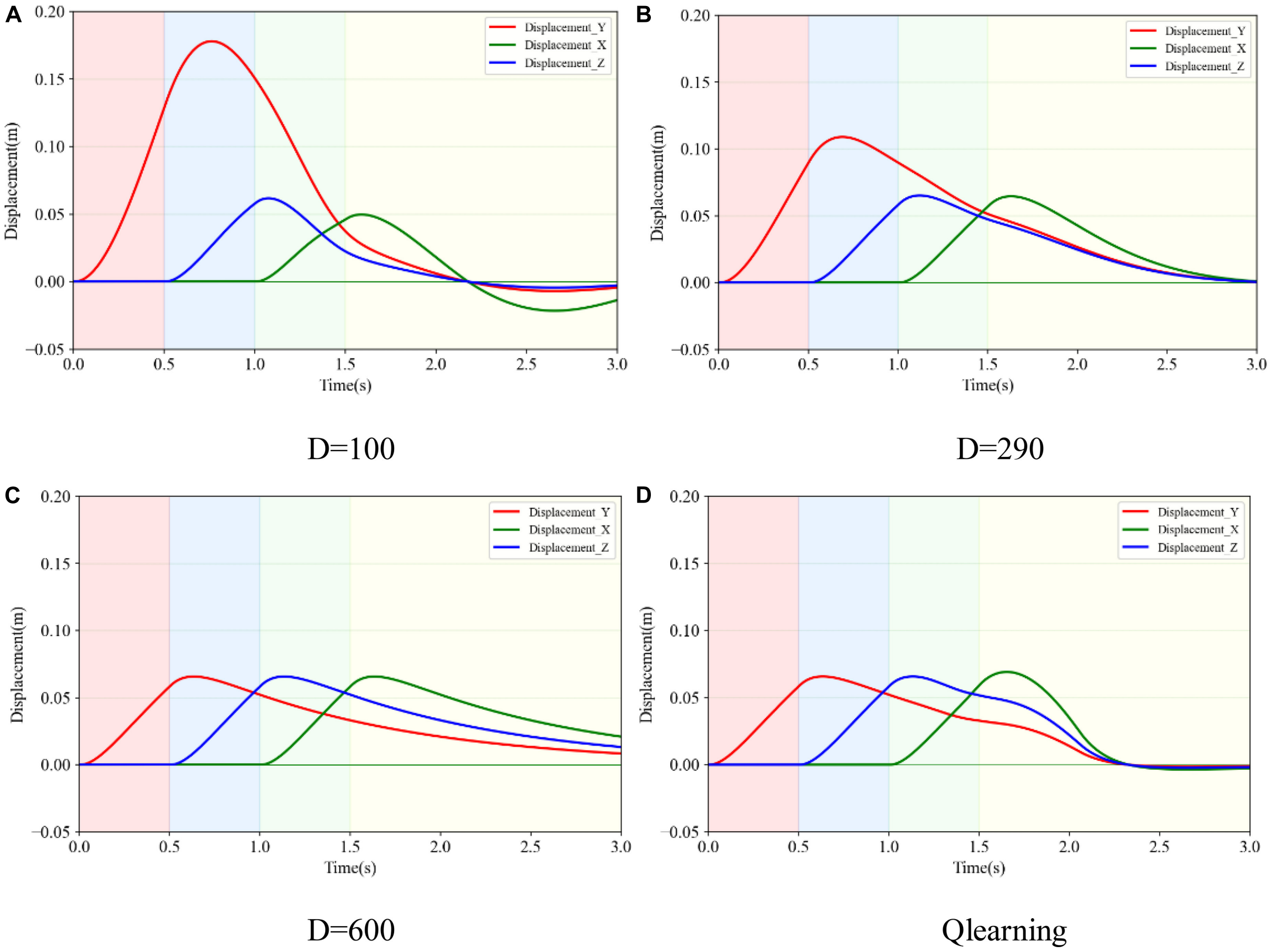


FIGURE 8
Three-dimensional displacement variance of the load subjected to three-dimensional impact in different damping cases. **(A)** Under damping case. **(B)** Critical damping case. **(C)** Over damping case. **(D)** Variable damping case based on Q-learning algorithm.

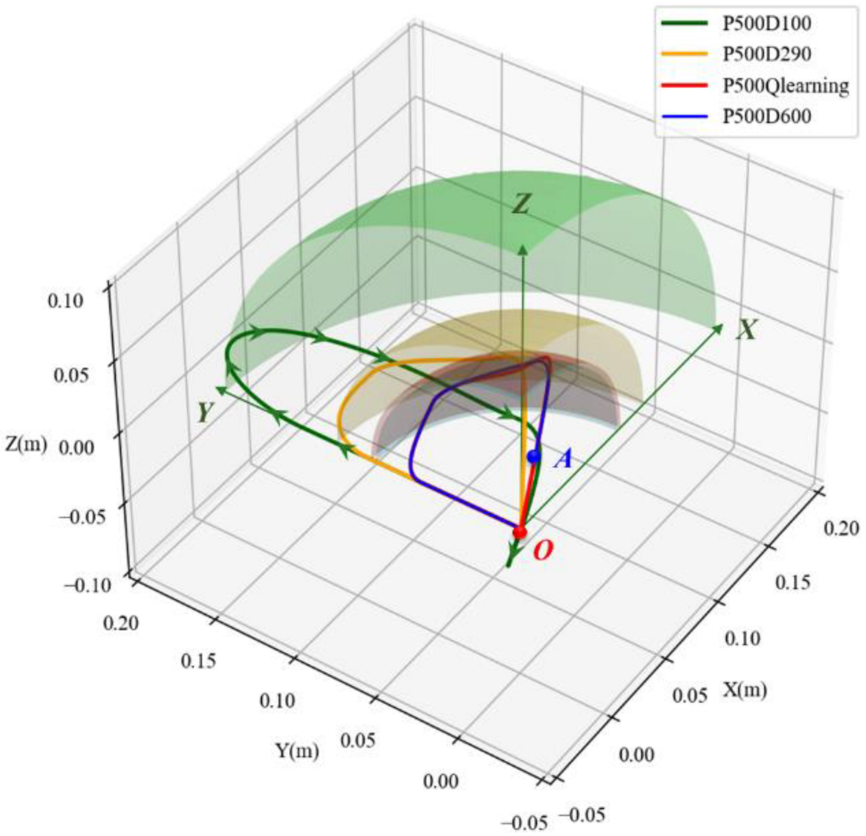


FIGURE 9
Comparison of spatial movement trajectories in different damping cases.

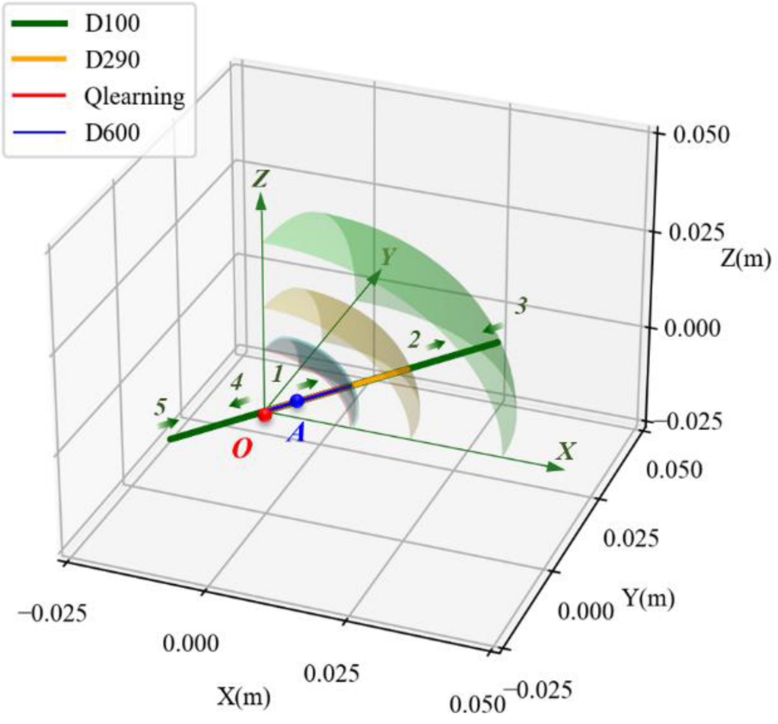


FIGURE 10
Comparison of motion trajectories of the robotic end in different damping cases.

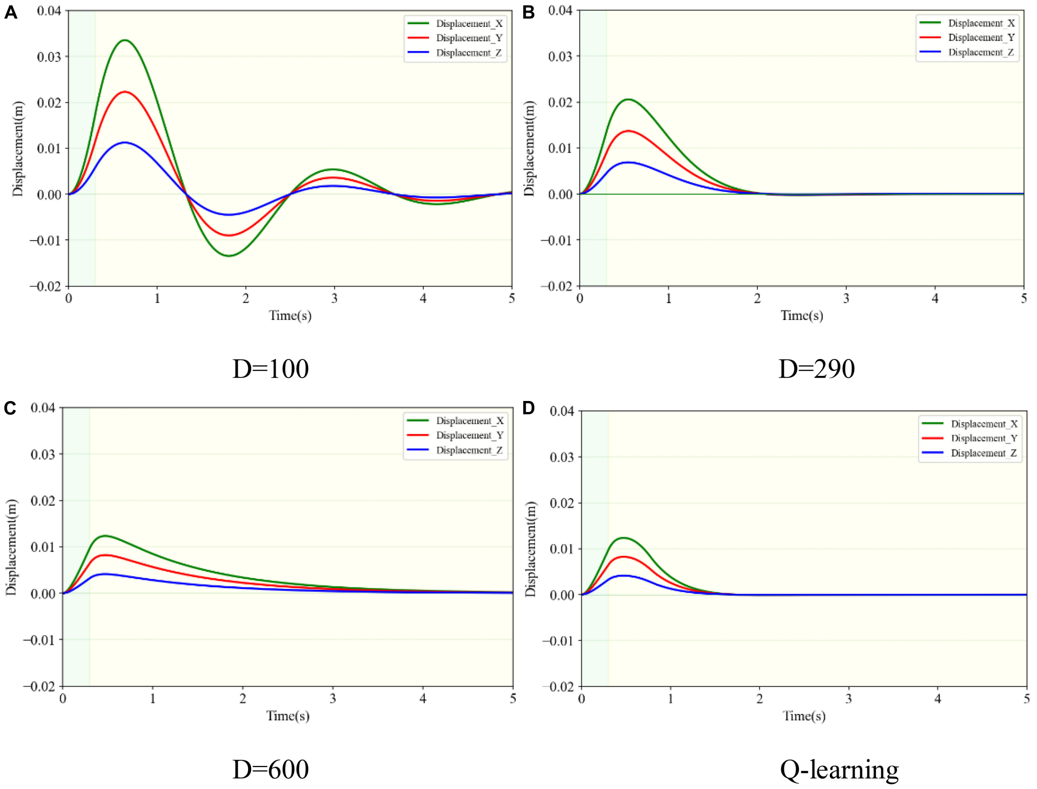


FIGURE 11
Three-dimensional displacement changes of the robotic end subjected to three-dimensional impact. **(A)** Under damping case. **(B)** Critical damping case. **(C)** Over damping case. **(D)** Variable damping case based on Q-learning algorithm.

prevent the system from oscillating. This paper focused on the impact force during a short period and recognized that the system was not subjected to a continuous force. When the damping term of the system persisted, the system eventually became stable.

3. Results

3.1. Reinforcement learning results

A simulation environment of the unconstrained load was built using the program Virtual Reality Educational Pathfinders (VREP) (Rohmer et al., 2013). The gravity acceleration in the vertical Z direction was set to zero to simulate the outer space environment. Taking the absolute coordinate system of the simulation environment as the reference coordinate system of the load, the real-time movement state could be obtained directly. In this simulation, the load mass was set to 64 kg, the impact force was set to 100N, and its duration time was 500 ms. The force was set to be along the positive direction of the *Y* axis, which acted on the load centroid. The training time for reinforcement learning was designed at 2.5 s so that the load could complete the whole process. The initial moment of the load was in a static state, then it moved in response to an external impact. The corresponding motion state was recorded in real time to obtain the current training state. The next action was selected according to the present state. The agent received a reward according to Equation 8 after each step. The total reward accumulated was recorded in one episode. The whole process was set at 3,000 training times.

The accumulated training reward of each episode is shown in **Figure 6**. The abscissa is the episode number, and the ordinate represents the total reward value obtained in each episode. According to Equation 8, the reward mechanism adopts a non-positive value so the total reward will be negative. Since the goal of reinforcement learning was to find the optimal strategy to maximize the cumulative reward value, the training performance improved as the cumulative reward value approached zero. According to **Figure 6**, it can be seen that the agent was an inexperienced individual during the first 750 training episodes. To learn the virtual restoring force control's damping coefficient and gain more experience, constant trial and error was required. Although the cumulative reward of each episode at this stage fluctuated significantly, it still showed an increasing trend in general. It could be proved that the agent gained some experience in training and the results moved in the right direction. After 750 episodes, the robot gradually learned the task target and the cumulative reward fluctuated slightly. Since the robot action selection strategy adopted the ε-greedy strategy, it enabled the agent with a certain degree of exploration ability. In this case, although the robot learned the action sequence leading to the task target, it still chose to explore a new action sequence with a certain small probability. It converged in the later stage of training and the cumulative reward value fluctuated slightly, which made no difference in the convergence of the whole training process.

The variable-damping controller based on reinforcement learning was tested and the results could be illustrated by the trajectory of the load after impact. The results were compared with fixed damping cases, as shown in **Figure 7**. The impact force was set as 100N and the duration time was set at 500ms. The stiffness

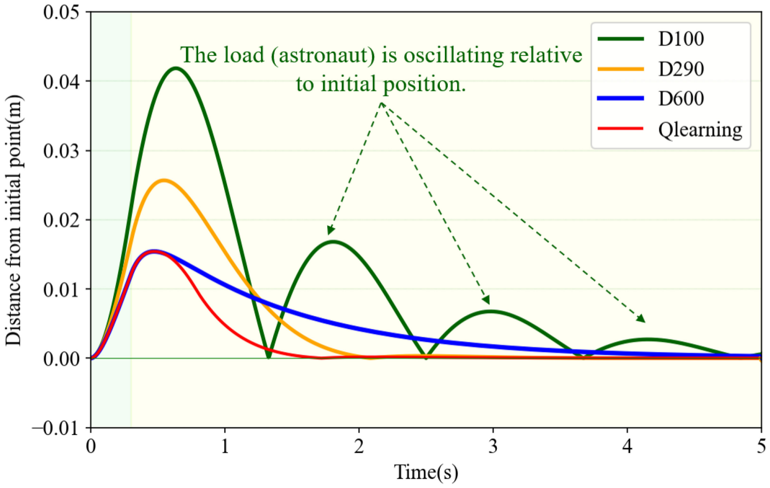


FIGURE 12
Distance from the initial point in different damping cases.

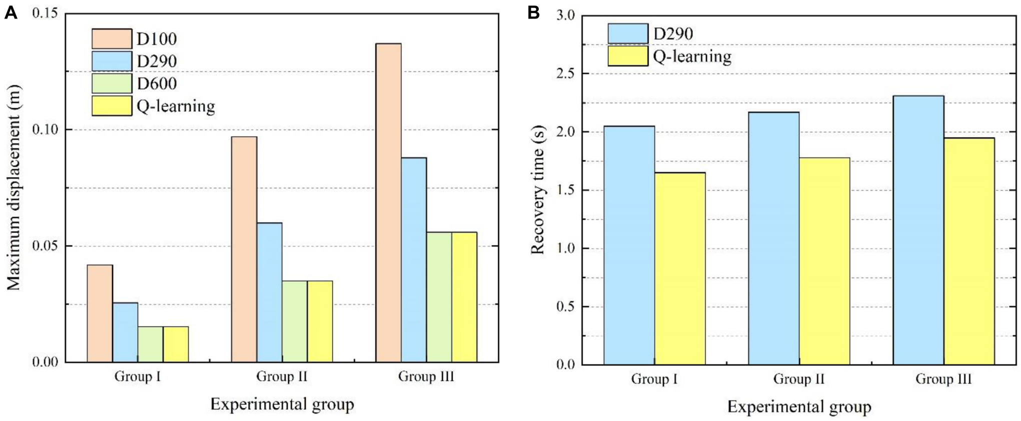


FIGURE 13
Comparisons of maximum displacement and recovery time for different experimental groups. **(A)** Comparison of maximum displacement. **(B)** Comparison of recovery time.

coefficient *K* was set at 500, and the fixed damping coefficient *D* was set at 100, 200, 290, 400, and 600, respectively. That is to say, there were five experiment groups to compare with the reinforcement learning result. As shown in Figure 7, when *D* was 290 as shown by the green solid line, the maximum displacement was 0.11 m and it could return to the initial position within 2.2 s. This value could be seen as the critical virtual damping of the load system. When *D* was 100 or 200, the system was in an underdamped state. It was in the overdamped state when *D* was 400 or 600. In the underdamped state, taking *D* = 100 as an example, shown by the light blue dotted line, the maximum displacement was 0.18 m, which was 0.07 m greater than the maximum displacement of critical damping. It could return to the initial position within 1.5 s. However, the load still had speed and failed to stop. It moved to the reverse maximum position and then moved back. In this way, the oscillation in relation to the initial position occurred repeatedly. Furthermore, it was unable to return to the initial position or stop within 2.5 s. When the load was in the overdamping state, taking D = 600 as an example, shown by the black dotted line, the maximum displacement after impact was 0.065 m, which was less than the maximum displacement of critical damping by 0.045 m and far less than the maximum displacement of under damping by 0.115 m. The load recovered very slowly because of the excessive damping. It moved towards the initial position during the recovery phase, but could not stop at the initial point within the specified time. There was still a position deviation of 0.015 m at 2.5 s.

The trajectory generated by the reinforcement learning algorithm is shown by the red solid line. The portion of the trajectory where the load started to deviate from its initial position after impact completely coincided with the overdamping case (*D* = 600), which indicates that the maximum damping was selected in the early stage to minimize displacement. When the impact weared off, the load began to return to its initial position after reaching maximum displacement under the virtual restoring force. For this process, the damping coefficient of reinforcement learning first decreased and then rose, so that the load could move towards the initial position quickly and try to stop at the initial point without overshooting. It can be seen from the red solid line that the load returned to the initial position within 1.7 s and finally remained stable, indicating the rapidity, stability, and recoverability of impact resistance. The variable damping control method took advantage of the small displacement deviation of the large damping case and the fast return of the small damping case. Compared with the critical damping case (D = 290) with better control effect in fixed damping, the maximum displacement of the reinforcement learning method reduced by 40.9% and the time to return to the original position shortened by 22.7%. Therefore, the variable damping control method met the requirements for impact resistance and pose maintenance.

3.2. Variable damping control results of end load

In order to evaluate the training results of reinforcement learning and solve the impact resistance problem subjected to three-dimensional impact, relative tests were carried out. According to Figure 4, the variable damping control method based on Q-learning was adopted for the dimension along the displacement direction, recorded as controller Y. The fixed damping control method was adopted for the dimension in vertical to displacement direction, recorded as controller X. Based on the orthogonal decoupling method for three-dimensional impact, four simulation experiments were designed. In these groups, the damping factor in the direction of vertical displacement X was set to a fixed value (D_x = 600), and the stiffness coefficient along the direction of displacement Y was set to 500. The damping coefficients were selected depending on the underdamping case, critical damping case, overdamping case, and variable-damping case. The corresponding values were recorded as 100, 290, 600, and Q-learning. The damping coefficient of the Q-learning method was variable. In these experiments, the velocity and displacement were not in the same straight line after the three-dimensional impact. Three components of the impact along YZX directions were continuously applied to the load within the first 1.5s. Each magnitude of the impact force was set at 100N and the duration time was 500ms. The results of different controllers were compared and analyzed.

Figure 8 indicates the displacement variance in XYZ directions after a three-dimensional impact. The solid red, blue, and green lines represent the trajectory changes in YZX directions, respectively. Taking Figure 8A as an example, only after the impact force was exerted in the appropriate direction did the corresponding displacement occur. At the starting time, the impact force was applied in the Y direction and the corresponding solid red line rose. The impact force in the Y direction disappeared after 0.5 s. At the time of 0.5 s, the impact force in the Z direction was exerted and disappeared after 0.5 s. The blue solid line kept rising. Similarly, the impact force in the X direction was applied during 1.0–1.5s, and the green solid line began to creep up. Comparing Figures 8A–D, it can be seen that the displacement change in the Y direction was the largest, of 0.18, 0.11, 0.065, and 0.065 m, respectively. They were consistent with the displacement change of the load after the unidirectional impact. The deviation from the initial position of the variable damping method after suffering an impact was the smallest. Comparing the X and Z directions, when in underdamping case (D = 100), there was an oscillation in the X direction. When in overdamping case (D = 600), it failed to return to the initial position in both X and Z directions within 3.0 s. When in the critical damping situation (D = 290), it took 3.0 s to return to the initial position. When in the variable damping case (Q-learning), it returned to the starting point within 2.3 s. The variable damping case took the least time to return.

The space motion trajectories generated by four experimental groups were compared, and the results are illustrated in Figure 9. Point O denotes the initial position of the load. The green straight line with an arrow represents the XYZ directions. The movement trajectories of underdamping, critical damping, overdamping, and variable damping control method are represented by green, yellow, blue, and red solid lines, respectively. It can be seen that the load was on point O at the initial time, and its motion trajectory was an irregular curve in space. From the perspective of Figure 9, the motion direction of the load was basically clockwise, according to the green arrows of the curve. The load first moved along the direction of increasing Y, then turned to the direction of increasing Z. After that the load moved to the direction of increasing X. Finally, it moveD back towards the origin point.

For each of the four motion trajectories, the maximum deviation values in relation to the initial position during the whole process was obtained. Taking this maximum displacement as the radius and the initial position as the center point of a sphere, the spheres with the maximal displacement under different controllers could be obtained. The maximal displacement spheres of the four groups are shown as the transparent surface, respectively, in Figure 9. The colors of these spheres are the same as their motion trajectories. For better comparison, only one-eighth of the maximal displacement sphere for the main motion space is shown. Comparing these transparent-colored surfaces, it could be observed that the smaller the damping factor was selected, the larger the sphere was. The other three groups of maximal displacement spheres were wrapped by the sphere (green transparent sphere) with the underdamping case (D = 100). The spheres of the overdamping case (D = 600) and variable damping case (Q-learning) almost coincided.

For the motion on the underdamping condition (D = 600), it could not stop immediately when the load returned to the initial position, according to its motion trajectory formed by the green solid line. However, it continued to move in the opposite direction through the origin for a certain distance and then returned. It resulted in oscillation in relation to the initial position. This corresponds to the part of the green solid line formed before the original point O. Combining the displacement curves in three directions in Figure 8A further supports the existence of oscillation.

For the two conditions of overdamping (D = 600) and variable damping (Q-learning), according to Figures 8C, D, the change magnitude and trend of the load-displacement in three directions of XYZ within 1.5s were basically the same. However, the load returned to its original position faster on the condition of variable damping. It could quickly return to the initial motion state within 2.3s and remain stable. In contrast, on the condition of overdamping, the load could not even return to the initial point within 3.0s. In terms of fast return, the performance with fixed large damping was not ideal.

The maximal displacement of load and the time taken to return to the original state after three-dimensional impact on four cases were comprehensively compared. The maximum displacement can serve as a good indicator of impact resistance stability. The impact resistance and stability will be greater and better as this index's value decreases. The time taken to recover to the initial state can be a good indicator of oscillation resistance. The less time required, the more quickly it will return to the initial state and the stronger its resilience will be. The maximal displacement on the variable damping condition (Q-learning) was 0.089m, which was basically consistent with the maximal displacement on the overdamping condition (D = 600), which was less than the maximal displacement of 0.18 m and 0.11m in the underdamping case (D = 100) and critical damping (D = 290) case. The values reduced by 50.6% and 19.1%, respectively, compared to the underdamping and critical damping cases. The performance of the variable damping controller was better. Moreover, the time taken to restore to the initial state under the variable damping condition (Q-learning) was 2.3 s, which was the least time consumed in the four groups. It was less than the counterparts on the critical damping condition (D = 290) and the overdamping condition (D = 600) with 3.1 and 5.2 s, which reduced by 25.8% and 55.8%. On the condition of underdamping (D = 100), the load could not return to its original position or remain stable after impact. In this case, its recovery performance was the worst. Therefore, according to the index comparison of the time taken to return to the initial state, the variable damping controller performed better in terms of a fast return. Based on the comparison results, it can be seen that the system showed the best stability, rapidity, and accuracy after suffering the impact under the variable damping method based on Q-learning, which verifies the feasibility and superiority of this method in impact resistance and position maintenance.

3.3. Variable damping control results of the robotic limbs

The simulation tests for the motion performance of the impact-affected end of the load were carried out in combination with the variable damping method and the dynamics of the robotic limb. The load was the same as in the above tests and connected to the robotic limb's end to form a system of manipulating the load. The load at the end of the robotic limb was subjected to external impact. The force acted on the load centroid, whose components in the *XYZ* axes were designed to be 30, 20, and 10 N. The duration time was set at 300 ms. Based on the simulation conditions, the motion performances of the robotic limb's end under four different damping controllers were compared. The experimental groups included three fixed damping cases (D = 100, 290, and 600) and one variable damping test based on the Q-learning algorithm. The four motion trajectories and the projections in the *XYZ* directions of the end load with respect to its initial position after impact are shown in Figures 10, 11.

The trajectories corresponding to the four simulation conditions were colored green, yellow, blue, and red. The origin O of the coordinate system in the figure represents the initial position of the end load, and the green straight lines with arrows represent the *XYZ* directions. They were consistent with the directions of the spatial absolute coordinate system. As shown in Figure 10, the end load started to move in the impact direction of the green arrows after the external impact force. Under the action of the restoring force, it moved toward the initial position after reaching the maximum displacement. Due to the underdamping case, the end load did not directly stop at the initial position. However, it moved past the initial position first and then returned, resulting in oscillation relative to the initial position. Its movement sequence is shown as the serial number from 1 to 5 in Figure 10. Yet on the other three conditions, the end load did not oscillate when receiving the impact force.

The maximal displacement of the end load under different conditions could be obtained in the same way as in section 3.2. The envelope surface of the maximum displacement of the end load is depicted in Figure 9 as the transparent surface. The maximal displacement in the underdamping case (D = 100) was 0.042 m, which was the largest. The counterpart in the variable damping case (Q-learning) was 0.015 m and it was the smallest. By contrast, the maximal displacement in the variable damping case was reduced by 64.3%. As shown in Figure 11, for the variable damping method, the load could restore to the initial state faster without oscillation, in a time of 1.65 s. However, for the condition of overdamping, the end load could not return to the initial position within 1.65s and only moved to point A. At this time, the distance between points O and A was 0.006 m, accounting for 40.0% of the maximal displacement in the whole process. Although the maximal displacement of the end load for the overdamping test was the least, its ability to return to the starting position was not strong. Compared with the underdamping and overdamping cases, the maximal displacement value and recovery time results of the critical damping case fell somewhere in between. For the case of underdamping, oscillation occurred and the load could not return to the initial position within 5 s, which was the maximal time designed for one single simulation episode. Therefore, it was considered that the recovery time was too long to meet the requirement for fast return, and the corresponding indicators were not compared. Thus, comparing the recovery time of the critical damping and variable damping cases, the former took 2.05 s and the latter only needed 1.65 s. The variable damping's recovery time was cut by 19.5%.

In order to compare the change of spatial distance with time between the real-time position and its initial point. Distance from the initial point under four different damping cases are shown in Figure 12, whose values were calculated by Equation 8. The last three conditions had a similar varying trend of distance, which first increased and then reduced to zero. However, in the underdamping case (D = 100), the trend changed periodically with amplitude attenuation. The load oscillated relative to its initial position on this condition. Compared to the other three cases, the variable damping method had the minimal deviation distance and the shortest return time. Furthermore, it could return directly to the initial position without oscillation.

In terms of resistance to impact at the end of the robotic limb, based on the above analysis, the end load with variable damping controller based on the Q-learning algorithm could quickly return to the initial position and stop after impact. During this process, the variable damping case had the least maximal displacement and minimal recovery time. It enabled the robotic limb to return fast and prevent oscillation.

In order to further verify the effectiveness of the proposed variable damping controller, experiments were carried out for different external impacts. For experimental group I, the force components in the *XYZ* axes were designed to be 30, 20, and 10 N. The duration time was set as 300 ms. For experimental group II, the force components in the *XYZ* axes were designed to be 50, 40, and 30 N. The duration time was set as 400 ms. For experimental group III, the force components in the *XYZ* axes were designed to be 50, 50, and 50 N. The duration time was set as 500 ms. Thus, the total impulse of external impact in the three experiments was 11.2, 28.3, and 43.3 Ns. Four different damping control methods were tested in each experimental group. The maximum displacement from the original point and recovery time in each case were emphatically compared and analyzed, as shown in Figure 13. As shown in Figure 13A, it can be seen that the system's maximum displacement was the least by the variable damping method for the three different impacts. Compared with the maximum displacement values to the underdamping and critical damping cases of all three experimental groups, the variable damping system's values were reduced by 39.3% and 62.1% on average. The underdamping system oscillated and the overdamping system could not stop within the specified time. Thus, for the recovery time, only the critical damping and variable damping were compared, as shown in **Figure 13B**. Compared with the critical damping method for the three different impacts, the variable damping method's return time was cut by 17.7% on average.

4. Discussion

This paper studied the issue of providing impact-resisting and position maintaining assistance for astronauts during EVA without the help of the space station's robotic arms. A wearable robotic limb system was introduced to give astronauts extra arms, which could help resist impact and maintain their position during EVA. The impact-resisting requirements for astronauts during EVA were analyzed. A variable damping controller based on the reinforcement learning algorithm was proposed. The combination system of an astronaut with robotic limbs was modeled and simplified. Compared with the fixed damping control method, the variable damping control method could meet all the impact-resisting requirements well by itself. It had better performance in preventing excessive deviation and exhibited fast return to the starting point. Meanwhile, it also had the capability of preventing oscillation and returning to the original position accurately. In the end, the appropriate simulation environment was built, and simulation experiments were conducted to confirm the method's rationality and viability.

However, there are still some limitations of the proposed method that will affect the performance in real-world situation. First, the weight of the astronaut and backpack was regarded as unchangeable in the simulation process. However, the actual situation is that for different astronauts, this value would slightly change. In order to improve the applicability of the method, this parameter also needs be taken as the input of the algorithm in further research. Second, the simulation environment was used for method validation, which is different from the real environment. It limits the experimental tests of the proposed method in practical application. In the future, it is necessary to set up a weightless experimental platform on the ground to simulate the outer space environment. We should let wearers with different weights carry out the relative tests to further verify the feasibility of the proposed method.

Data availability statement

The original contributions presented in this study are included in the article/**Supplementary material**, further inquiries can be directed to the corresponding author.

Author contributions

SZ and YZ proposed the concept. SZ, TZ, and DS established the algorithm model and carried out the simulation experiments under the supervision of JZ and YZ. SZ wrote the original manuscript. TZ and DS helped review the manuscript. All authors read and agreed on the final version of the manuscript.

Funding

This research was funded by the National Natural Science Foundation of China (NSFC) (Nos. 52025054 and 52105016).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2023.1093718/full#supplementary-material

References

Ackerman, E. (2019). *Skybot F-850 will spend a week on the ISS charming astronauts with its sense of humor*. Available online at: https://spectrum.ieee.org/russian-humanoid-robot-to-pilot-soyuz-capsule-to-iss-this-week (accessed November 1, 2022).

Chang, Y.-W. (2015). The first decade of commercial space tourism. *Acta Astronaut.* 108, 79–91. doi: 10.1016/j.actaastro.2014.12.004

Chen, L., Jiang, Z., Cheng, L., Knoll, A. C., and Zhou, M. (2022). Deep reinforcement learning based trajectory planning under uncertain constraints. *Front. Neurorobot.* 16:883562. doi: 10.3389/fnbot.2022.883562

Chien, S., and Wagstaff, K. L. (2017). Robotic space exploration agents. *Sci. Robot.* 2:4831. doi: 10.1126/scirobotics.aan4831

Clifton, J., and Laber, E. (2020). Q-Learning: Theory and applications. *Ann. Rev. Stat. Appl.* 7, 279–301. doi: 10.1146/annurev-statistics-031219-041220

Cong, L., Liang, H., Ruppel, P., Shi, Y., Gorner, M., Hendrich, N., et al. (2022). Reinforcement learning with vision-proprioception model for robot planar pushing. *Front. Neurorobot.* 16:829437. doi: 10.3389/fnbot.2022.829437

Diftler, M. A., Mehling, J. S., and Abdallah, M. E. (2011). "Robonaut 2 – the first humanoid robot in space," in *Proceedings of the 2011 IEEE international conference on robotics and automation*, (Shanghai: IEEE). doi: 10.1109/ICRA.2011.5979830

Flores-Abad, A., Ma, O., Pham, K., and Ulrich, S. (2014). A review of space robotics technologies for on-orbit servicing. *Prog. Aerosp. Sci.* 68, 1–26. doi: 10.1016/j.paerosci.2014.03.002

Garcia, M. (2019). *NASA spacewalker Anne McClain*. Available online at: https://www.nasa.gov/image-feature/nasa-spacewalker-anne-mcclain (accessed November 1, 2022).

Hall, L. (2013). *NASA's ironman-like exoskeleton could give astronauts, paraplegics improved mobility and strength*. Available online at: https://www.nasa.gov/offices/oct/home/feature_exoskeleton.html (accessed November 1, 2022).

Hutabarat, Y., Ekkachai, K., Hayashibe, M., and Kongprawechnon, W. (2020). Reinforcement Q-learning control with reward shaping function for swing phase control in a semi-active prosthetic knee. *Front. Neurorobot.* 14:565702. doi: 10.3389/fnbot.2020.565702

Jacobstein, N., Bellingham, J., and Yang, G.-Z. (2017). Robotics for space and marine sciences. *Sci. Robot.* 2:5594. doi: 10.1126/scirobotics.aan5594

Jiang, H., Hawkes, E. W., Fuller, C., Estrada, M. A., and Suresh, S. A. (2017). A robotic device using gecko-inspired adhesives can grasp and manipulate large objects in microgravity. *Sci. Robot.* 2:4545. doi: 10.1126/scirobotics.aan4545

Lester, D. F., Hodges, K. V., and Anderson, R. C. (2017). Exploration telepresence a strategy for optimizing scientific research at remote space destinations. *Sci. Robot.* 2:4383. doi: 10.1126/scirobotics.aan4383

Li, Y., Li, D., Zhu, W., Sun, J., Zhang, X., and Li, S. (2022). Constrained motion planning of 7-DOF space manipulator via deep reinforcement learning combined with artificial potential field. *Aerospace* 9:163. doi: 10.3390/aerospace9030163

Liu, Y., Jiang, D., Yun, J., Sun, Y., Li, C., Jiang, G., et al. (2021). Self-tuning control of manipulator positioning based on fuzzy PID and PSO algorithm. *Front. Bioeng. Biotechnol.* 9:817723. doi: 10.3389/fbioe.2021.817723

McHenry, N., Davis, L., Gomez, I., Coute, N., Roehrs, N., Villagran, C., et al. (2020). "Design of an AR visor display system for extravehicular activity operations," in *Proceedings of the 2020 IEEE aerospace conference*, Big Sky, MT.

Mohon, L. (2014). *STS-112 spacewalk*. Available online at: https://www.nasa.gov/centers/marshall/history/launch_of_sts-112.html (accessed November 1, 2022).

Nokleby, S. B. (2007). Singularity analysis of the Canadarm2. *Mech. Mach. Theory* 42, 442–454. doi: 10.1016/j.mechmachtheory.2006.04.004

Ohnishi, S., Uchibe, E., Yamaguchi, Y., Nakanishi, K., Yasui, Y., and Ishii, S. (2019). Constrained deep Q-learning gradually approaching ordinary q-learning. *Front. Neurorobot.* 13:103. doi: 10.3389/fnbot.2019.00103

Olivieri, L., Brunello, A., Sarego, G., Valmorbida, A., and Lorenzini, E. C. (2021). An in-line damper for tethers-in-space oscillations dissipation. *Acta Astronaut.* 189, 559–566. doi: 10.1016/j.actaastro.2021.09.012

Post, M. A., Yan, X.-T., and Letier, P. (2021). Modularity for the future in space robotics: A review. *Acta Astronaut.* 189, 530–547. doi: 10.1016/j.actaastro.2021.09.007

Raina, D., Gora, S., Maheshwari, D., and Shah, S. V. (2021). Impact modeling and reactionless control for post-capturing and maneuvering of orbiting objects using a multi-arm space robot. *Acta Astronaut.* 182, 21–36. doi: 10.1016/j.actaastro.2021.01.034

Rohmer, E., Singh, S. P. N., and Freese, M. (2013). "V-REP: A versatile and scalable robot simulation framework," in *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*, Tokyo.

Ruttley, T. M., Robinson, J. A., and Gerstenmaier, W. H. (2017). The international space station: Collaboration, utilization, and commercialization*. *Soc. Sci. Q.* 98, 1160–1174. doi: 10.1111/ssqu.12469

Su, Y., Hou, X., Li, L., Cao, G., Chen, X., Jin, T., et al. (2020). Study on impact energy absorption and adhesion of biomimetic buffer system for space robots. *Adv. Space Res.* 65, 1353–1366. doi: 10.1016/j.asr.2019.12.006

Wang, S., Huang, L., Jiang, D., Sun, Y., Jiang, G., Li, J., et al. (2022). Improved multi-stream convolutional block attention module for sEMG-based gesture recognition. *Front. Bioeng. Biotechnol.* 10:909023. doi: 10.3389/fbioe.2022.909023

Wang, X., Xu, B., Cheng, Y., Wang, H., and Sun, F. (2022). Robust adaptive learning control of space robot for target capturing using neural network. *IEEE Trans. Neural Netw. Learn. Syst.* doi: 10.1109/TNNLS.2022.3144569

Webber, D. (2013). Space tourism: Its history, future and importance. *Acta Astronaut.* 92, 138–143. doi: 10.1016/j.actaastro.2012.04.038

Zhang, X., Liu, J., Feng, J., Liu, Y., and Ju, Z. (2020a). Effective capture of nongraspable objects for space robots using geometric cage pairs. *IEEE/ASME Trans. Mechatron.* 25, 95–107. doi: 10.1109/tmech.2019.2952552

Zhang, X., Liu, J., Gao, Q., and Ju, Z. (2020b). Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. *Nonlinear Dyn.* 100, 2449–2467. doi: 10.1007/s11071-020-05615-5

Zhang, X., Xiao, F., Tong, X., Yun, J., Liu, Y., Sun, Y., et al. (2022). Time optimal trajectory planning based on improved sparrow search algorithm. *Front. Bioeng. Biotechnol.* 10:852408. doi: 10.3389/fbioe.2022.852408

Zhao, G., Jiang, D., Liu, X., Tong, X., Sun, Y., Tao, B., et al. (2022). A Tandem robotic arm inverse kinematic solution based on an improved particle swarm algorithm. *Front. Bioeng. Biotechnol.* 10:832829. doi: 10.3389/fbioe.2022.832829

Zhao, S., Zhao, J., Sui, D., Wang, T., Zheng, T., Zhao, C., et al. (2021). Modular robotic limbs for astronaut activities assistance. *Sensors (Basel)* 21:6305. doi: 10.3390/s21186305

Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H. (2007). Evolved and designed self-reproducing modular robotics. *IEEE Trans. Robot.* 23, 308–319. doi: 10.1109/tro.2007.894685

TYPE Original Research
PUBLISHED 24 February 2023
DOI 10.3389/fnbot.2023.1093132

OPEN ACCESS

EDITED BY
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

REVIEWED BY
Huaming Qian,
Harbin Engineering University, China
Tim Verbelen,
Ghent University, Belgium
Vittorio Cuculo,
University of Milan, Italy

*CORRESPONDENCE
Feng Zhu
✉ fzhu@sia.cn

RECEIVED 08 November 2022
ACCEPTED 08 February 2023
PUBLISHED 24 February 2023

CITATION
Sun H, Zhu F, Li Y, Zhao P, Kong Y, Wang J, Wan Y and Fu S (2023) Viewpoint planning with transition management for active object recognition. *Front. Neurorobot.* 17:1093132. doi: 10.3389/fnbot.2023.1093132

Viewpoint planning with transition management for active object recognition

Haibo Sun[1,2,3,4], Feng Zhu[2,3,4]*, Yangyang Li[2,3,4,5], Pengfei Zhao[2,3,4,5], Yanzi Kong[2,3,4,5], Jianyu Wang[1,2,3,4], Yingcai Wan[1] and Shuangfei Fu[2,3,4]

[1]Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China, [2]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang, China, [3]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, [4]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China, [5]University of Chinese Academy of Sciences, Beijing, China

Active object recognition (AOR) provides a paradigm where an agent can capture additional evidence by purposefully changing its viewpoint to improve the quality of recognition. One of the most concerned problems in AOR is viewpoint planning (VP) which refers to developing a policy to determine the next viewpoints of the agent. A research trend is to solve the VP problem with reinforcement learning, namely to use the viewpoint transitions explored by the agent to train the VP policy. However, most research discards the trained transitions, which may lead to an inefficient use of the explored transitions. To solve this challenge, we present a novel VP method with transition management based on reinforcement learning, which can reuse the explored viewpoint transitions. To be specific, a learning framework of the VP policy is first established *via* the deterministic policy gradient theory, which provides an opportunity to reuse the explored transitions. Then, we design a scheme of viewpoint transition management that can store the explored transitions and decide which transitions are used for the policy learning. Finally, within the framework, we develop an algorithm based on twin delayed deep deterministic policy gradient and the designed scheme to train the VP policy. Experiments on the public and challenging dataset GERMS show the effectiveness of our method in comparison with several competing approaches.

KEYWORDS

active object recognition, viewpoint planning, deterministic policy gradient, twin delayed deep deterministic policy gradient, viewpoint transition management, reinforcement learning

1. Introduction

Visual object recognition has a wide range of applications e.g., automatic driving (Behl et al., 2017), robotics (Stria and Hlavác, 2018), medical diagnostic (Duan et al., 2019), environmental perception (Roynard et al., 2018), etc. Most recognition systems merely take a single viewpoint image as input and produce a category label estimate as output (Jayaraman and Grauman, 2019). It is prone to the recognition errors when the image can not provide sufficient information. In contrast, the visual behavior of people is an active process so as to more clearly perceive their surroundings. As shown in Figure 1, in daily life, people can intelligently observe an object from different viewpoints to determine the identity of the object. Similarly, if the viewpoint of an agent can be adjusted (e.g., mobile robots and

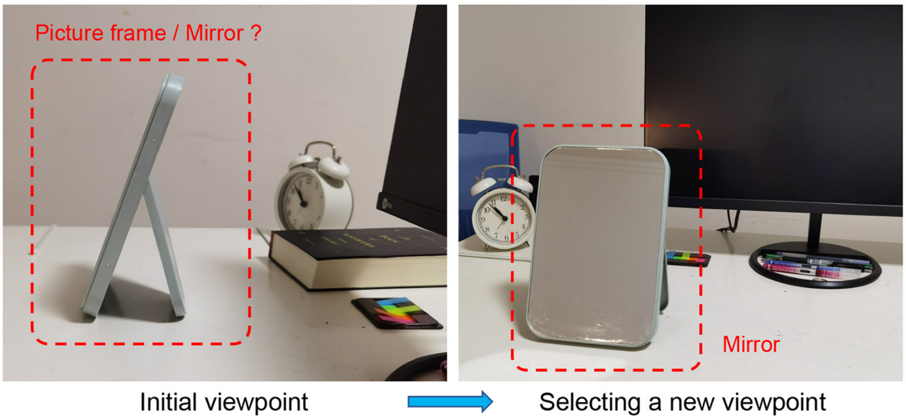


FIGURE 1
An example illustrating the active preception process of people.

autonomous vehicles), more valuable information will be obtained to boost the recognition performance.

As a branch of active vision (Parr et al., 2021), active object recognition (AOR) (Patten et al., 2015; Wu et al., 2015; Potthast et al., 2016; Van de Maele et al., 2022) is a typical technology to realize the above idea, which aims to collect additional clues by purposefully changing the viewpoint of an agent to improve the quality of recognition. Andreopoulos and Tsotsos (2013) and Zeng et al. (2020) review a series of classical AOR methods. One of the most concerned problems in AOR is viewpoint planning (VP) that refers to developing a policy to determine the next viewpoints of the agent. In recent years, researchers mainly focus on using reinforcement learning to solve the VP problem (Becerra et al., 2014; Malmir et al., 2015; Malmir and Cottrell, 2017; Liu et al., 2018a), namely to use the viewpoint transitions explored by the agent to train the VP policy. Becerra et al. (2014) formally define object recognition as a partially observable Markov decision process problem and uses stochastic dynamic programming to address the problem. As a pioneering work, Malmir et al. (2015) provide a public AOR dataset called GERMS that includes 136 objects with different view images and develops a deep Q-learning (DQL) system to learn to actively verify objects by using standard back-propagation and Q-learning. In the same way, Liu et al. (2018a) design a hierarchical local-receptive-field architecture to predict object label and learns a VP policy by combining extreme learning machine and Q-learning. Similar to Becerra et al. (2014), AOR is also modeled as a partially observable Markov decision process by Malmir and Cottrell (2017). The difference is that a belief tree search is built to find near-optimal action values which correspond to the next best viewpoints. These VP methods explore discrete viewpoint space, which may introduce significant quantization errors. Hence, Liu et al. (2018b) present a continuous VP method based on trust region policy optimization (TRPO) (Schulman et al., 2015) and adopts extreme learning machine (Huang et al., 2006) to reduce computational complexity. It shows a promising result on the GERMS dataset compared to the discrete VP methods. However, due to the on-policy characteristic of TRPO, the trained viewpoint transitions will be discarded by the agent, which may lead to an inefficient use of the explored transitions.

The deterministic policy gradient theory (Silver et al., 2014) is proposed for reinforcement learning with continuous actions and introduces an off-policy actor-critic algorithm (OPDAC-Q) to learn a deterministic target policy. Lillicrap et al. (2015) present a deep deterministic policy gradient (DDPG) approach that combines deterministic policy gradient with DQN (Mnih et al., 2013, 2015) to learn policies in high-dimensional continuous action spaces. Fujimoto et al. (2018) contribute a mechanism that takes the minimum value between a pair of critics in the actor-critic algorithm of Silver et al. (2014) to tackle the function approximation errors. The deterministic policy gradient theory has been widely applied in various fields, such as electricity market (Liang et al., 2020), vehicle speed tracking control (Hao et al., 2021), fuzzy PID controller (Shi et al., 2020), quadrotor control (Wang et al., 2020), energy efficiency (Zhang et al., 2020), and autonomous underwater vehicles (Sun et al., 2020; Wu et al., 2022). However, to our best knowledge, it has never been employed in the AOR task.

In this work, we present a novel continuous VP method with transition management based on reinforcement learning. This method can efficiently use the explored viewpoint transitions to learn the continuous VP policy. Concretely, a learning framework of the continuous VP policy is established using the deterministic policy gradient theory, which provides an opportunity to reuse the explored transitions owing to the off-policy characteristic of the theory. Then, we design a scheme of viewpoint transition management that can store the explored transitions and decide which transitions are used for the policy learning. The scheme is implemented by introducing and improving the prioritized experience replay technology (Schaul et al., 2016). The improvements include: (1) We improve the estimation approach of temporal difference (TD) error with the clipped double Q-learning algorithm (Fujimoto et al., 2018) so as to adapt to our continuous VP framework. (2) We utilize importance-sampling to correct the estimation bias of TD error produced by the prioritized replay. Finally, within the framework, we develop an algorithm based on twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the designed scheme to train the continuous VP policy. Experimental results on the public dataset GERMS demonstrate the effectiveness of the proposed VP method.

The key contributions of this work are

- A novel continuous VP method with transition management for AOR is presented to solve the problem of inefficient use of the explored viewpoint transitions in the existing continuous VP method.
- We establish a learning framework of the continuous VP policy *via* the deterministic policy gradient theory.
- A scheme of viewpoint transition management is designed, which is implemented by introducing and improving the prioritized experience replay technology.
- We develop an algorithm based on twin delayed deep deterministic policy gradient and the designed scheme to train the continuous VP policy.

The rest of this paper is structured as follows: Section 2 formulates the VP problem. Section 3 details the proposed framework for the solution of the problem. Finally, the implementation and experimental results, as well as conclusions are further provided in Sections 4, 5.

2. Problem definition

An AOR system mounted on an automatic mobile agent allows the agent to identify an object by dealing with the images captured from different viewpoints. Suppose at the initial time $t = 0$, an object to be identified is given from an object library containing M objects and the agent captures an image I_{Φ_0} from the initial viewpoint Φ_0. The classifier $\mathcal{C}(\cdot)$ in the AOR system will give a probability prediction $\mathcal{C}(I_{\Phi_0})$ of the object according to the image I_{Φ_0}. $\mathcal{C}(I_{\Phi_0})$ is a M dimensional vector where every element denotes recognition probability of different objects in the library. When the prediction is uncertain [i.e., the maximum probability in $\mathcal{C}(I_{\Phi_0})$ is less than the preset threshold], the agent will move to explore more viewpoints to improve recognition performance. This requires the system plans a relative movement action a_t for the agent to obtain a new viewpoint $\Phi_{t+1} = \Phi_t + a_t$. The new image $I_{\Phi_{t+1}}$ captured from the viewpoint Φ_{t+1} will be used for the recognition again. This process is repeated several times until a stop condition (e.g., planning up to T_{max} time steps or reaching the preset probability threshold) is reached.

An undesirable planning action may make it difficult for the agent to capture useful images for recognition. Therefore, we need to find an effective VP policy for the AOR system. For this purpose, the VP problem is considered as a reinforcement learning paradigm which can be formulated as a Markov decision process. The process is described with a six-element tuple $< S, A, r, \mathcal{P}, \gamma, u >$.

- S represents a set of continuous states in which each state s is produced by the predictions of corresponding images captured from different viewpoints.
- A is a set of continuous actions which are determined by the agent. Each action a in the set is used for the agent to get a new viewpoint.
- $r : S{\times}A{\rightarrow}\mathbb{R}$ is a reward function designed to evaluate the quality of selecting a viewpoint.
- $\mathcal{P} : S{\times}A{\times}S{\rightarrow}[0, 1]$ denotes the transition probability. It describes the possibility of transferring to the subsequent state s' after the action a is selected in the state s.
- $\gamma \in [0, 1]$ is a discount factor used to adjust the attention between present and future rewards.
- $u : S{\rightarrow}A$ is a deterministic continuous VP policy [i.e., $a = u(s)$] that can generate an action for the agent to get a new viewpoint in a certain state.

The VP problem is transformed to solve the optimal policy u^* in the setting of reinforcement learning.

3. Method

3.1. Overview

In reinforcement learning, the optimal policy u^* can be achieved by maximizing the expected return over all episodes. At any time step t of each episode, with a given state $s_t{\in}S$, the agent plans an action $a_t{\in}A$ according to its current policy u ($a_t = u(s_t)$), receiving a reward $r(s_t, a_t)$ and the new state $s_{t+1}{\sim}\mathcal{P}(s_{t+1}|s_t, a_t)$. ($(s_t, a_t, r_t, s_{t+1})$ is called the viewpoint transition in the AOR task.) The return is defined as the cumulative discounted reward $\sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$ where T is the end time step of planning. Let $Q^u(s_t, a_t)$ be the expected return when performing action a_t in state s_t under the policy u. $Q^u(s_t, a_t)$ is defined as

$$Q^u(s_t, a_t) = \mathop{\mathbb{E}}_{s_{t+1}\sim\mathcal{P}(s_{t+1}|s_t,a_t)}[\sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)|s_t, a_t] \tag{1}$$

which is known as the action value function. u^* can be solved by maximizing the expected value of Equation (1) over the whole state space

$$u^* = \max_u \mathbb{E}_{s_t\sim d(\cdot)}[Q^u(s_t, a_t)|a_t = u(s_t)] \tag{2}$$

where $d(\cdot)$ is the state probability density of Markov decision process in steady state distribution (Bellemare et al., 2017).

We assume the deterministic continuous VP policy u is parameterized by θ and denote it as $u(s; \theta)$. Naturally, Equation (2) can be transformed to an optimization with respect to θ that maximize the objective

$$J(\theta) = \mathbb{E}_{s_t\sim d(\cdot)}[Q^u(s_t, a_t)|a_t = u(s_t; \theta)]. \tag{3}$$

To solve the optimization of Equation (3), the deterministic policy gradient theory (Silver et al., 2014) is introduced to iteratively update the parameters θ by taking the gradient of Equation (3)

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_t\sim d(\cdot)}[\nabla_\theta u(s_t; \theta) \nabla_a Q^u(s_t, a_t)|a_t = u(s_t; \theta)]. \tag{4}$$

We utilize (Equation 4) as a framework to learn the optimal deterministic continuous VP policy $u(s_t; \theta^*)$ for AOR. The reason why this framework can reuse the explored viewpoint transitions is the off-policy characteristic of the deterministic policy gradient

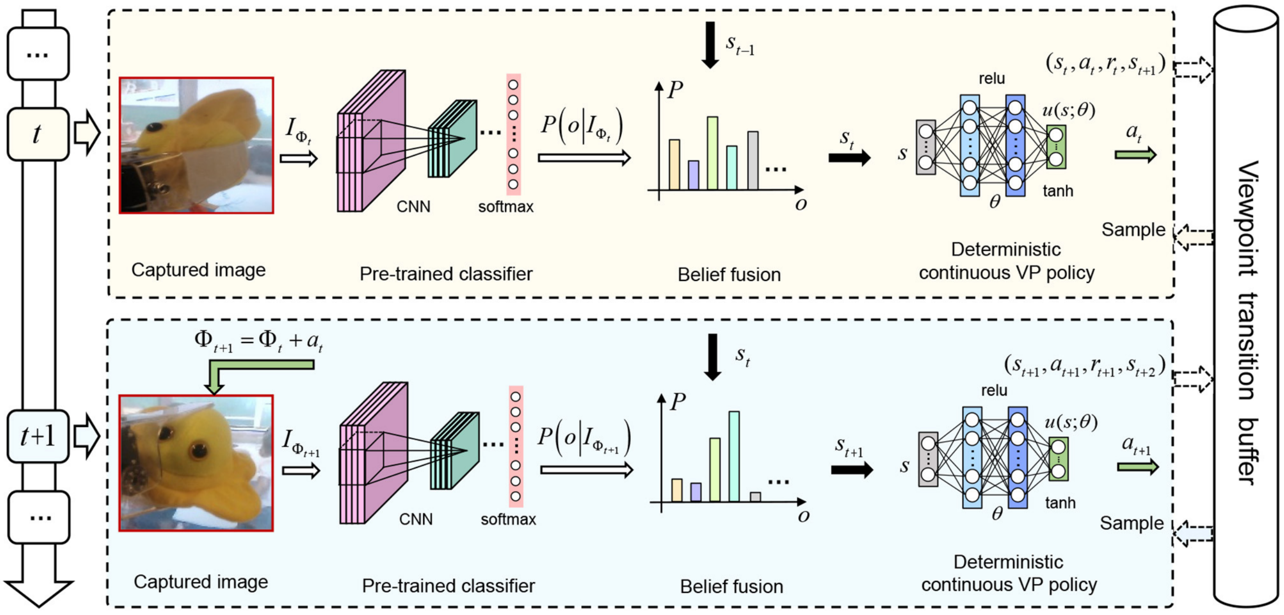


FIGURE 2
The pipeline of active object recognition based on deterministic continuous viewpoint planning. The deterministic policy gradient theory (Silver et al., 2014) is introduced to build a framework of continuous viewpoint planning. We design a scheme of viewpoint transition management to store and replay the explored viewpoint transitions. Within the framework, we develop an algorithm based on TD3 (Fujimoto et al., 2018) and the scheme to train the VP policy network. During the training, the agent stores the explored viewpoint transition (s_t, a_t, r_t, s_{t+1}) in the viewpoint transition buffer and samples a mini-batch transitions from it to train the VP policy network at each time step.

theory, i.e., the viewpoint transitions explored by any policy can be used for the calculation of the gradient in Equation (4), because the gradient is only related to the distribution of state s_t (Silver et al., 2014). The pipeline of our AOR is shown in Figure 2 where the VP policy $u(s_t; \theta)$ is represented by a three-layer fully-connected neural network with the parameters θ. The policy network $u(s_t; \theta)$ takes a state s_t as input and outputs a deterministic action $a_t = u(s_t; \theta)$. In the following, the representations of state s_t and reward function $r(s_t, a_t)$ will be elaborated. Additionally, we will design a scheme of viewpoint transition management and develop a training algorithm based on twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the scheme for the learning of $u(s_t; \theta^*)$ within the framework.

3.2. Recognition state

As shown in Figure 2, we first use a convolutional neural network (CNN) model to extract features from the captured image I_{Φ_t} and then recognize the concerned objects with a *softmax* layer added the top of the CNN model. The CNN model and the *softmax* layer constitute a classifier $\mathcal{C}(\cdot)$ which is pre-trained with the images from different viewpoints of the concerned objects. The parameters of the classifier are fixed when training the VP policy network. The classifier outputs a belief vector $\mathcal{C}(I_{\Phi_t})$ where every element denotes recognition probability of different objects. The *oth* element in the vector is represented as $P(o|I_{\Phi_t})$ where $o = 1, 2, ..., M$ is the object label. The recognition state s_t is a posterior probability distribution over different objects at time step t, which is produced by the captured images. It is also expressed as a vector where the *oth* element is $P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t}), o = 1, 2, ..., M$. According to naive Bayes (Paletta and Pinz, 2000), $P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t})$ is given as

$$\xi_t P(o|I_{\Phi_t}) P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_{t-1}}) \tag{5}$$

where ξ_t is a normalizing coefficient.

3.3. Reward function

Reward function $r(s_t, a_t)$ (denoted as r_t for simplicity) is used to evaluate the quality of selecting a viewpoint. As described in Section 3.2, state is a posterior probability distribution over different objects. The flatter the distribution is, the stronger the recognition uncertainty is. To quantify the uncertainty, information entropy (Zhao et al., 2016; Liu et al., 2018b) is utilized and the uncertainty in state s_t is denoted as $H(s_t) = -\sum_o P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t}) \log P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t})$. The purpose of AOR is to reduce the uncertainty of recognition through viewpoint planning. Therefore, we can design the reward function according to the change of uncertainty before and after viewpoint selection. The resulting reward function is

$$r_t = \begin{cases} -1, & \hat{o}_{t+1} \neq o^* \\ 0, & \hat{o}_{t+1} = o^*, H(s_{t+1}) \geq H(s_t) \\ 1, & \hat{o}_{t+1} = o^*, H(s_{t+1}) < H(s_t) \end{cases} \tag{6}$$

where o^* is the object label and $\hat{o}_{t+1} = argmax_o P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_{t+1}})$ is the predicted result. When the predicted result is right ($\hat{o}_{t+1} = o^*$) and the uncertainty is reduced ($H(s_{t+1}) < H(s_t)$), it indicates that this viewpoint selection

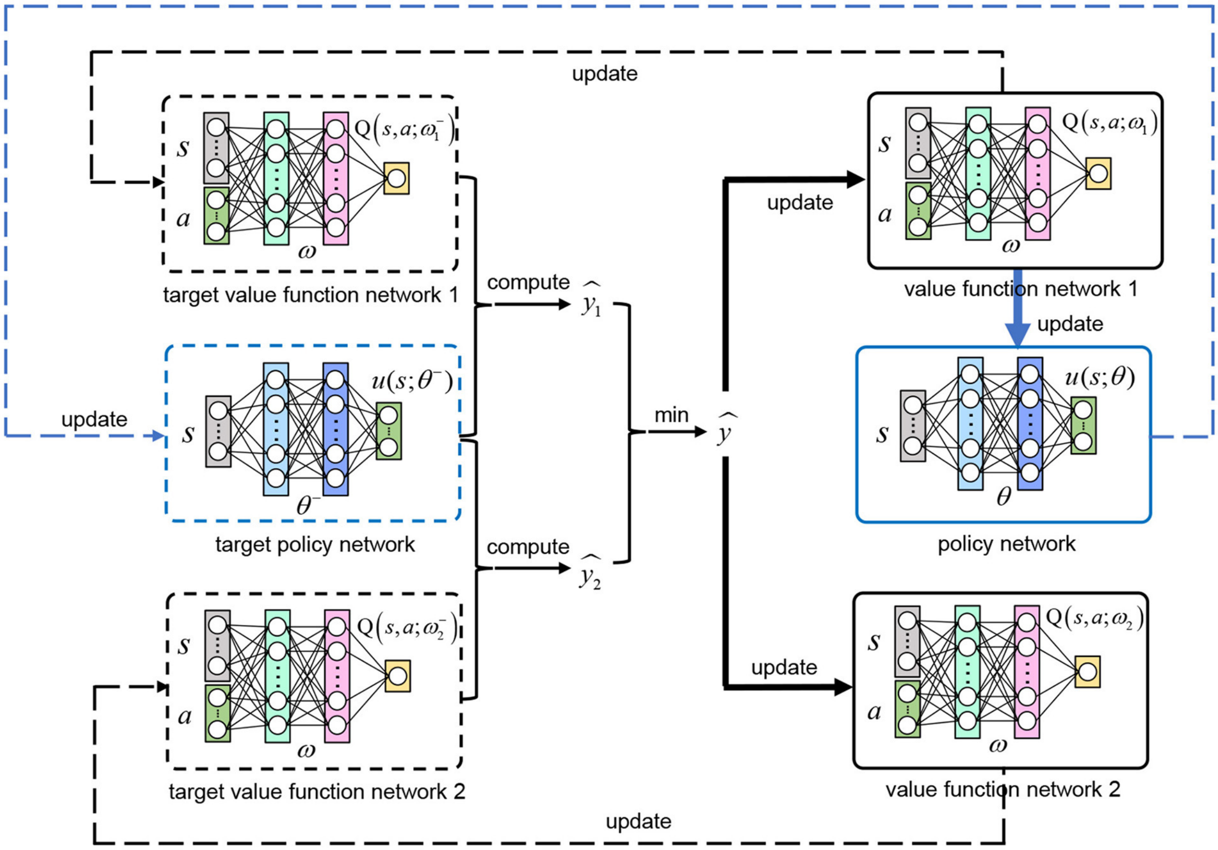


FIGURE 3
The relationship between the six networks. The TD target $\hat{y}$ is estimated with the target value function network 1 and 2 using our clipped double Q-learning and bias correction based algorithm (Equation 12), which is used to update the value function network 1 and 2. With the gradient of $Q(s_t, a_t; \omega_1)$ to a, the policy network is updated with Equation (13). Three target networks ($u(s_t; \theta^-)$, $Q(s_t, a_t; \omega_1^-)$, $Q(s_t, a_t; \omega_2^-)$) adopt soft updates according to their corresponding evaluation networks ($u(s_t; \theta)$, $Q(s_t, a_t; \omega_1)$, $Q(s_t, a_t; \omega_2)$).

is valuable for recognition. On the contrary, other situations mean that this viewpoint selection is not good.

3.4. Viewpoint transition management

The agent can obtain a transition (s_t, a_t, r_t, s_{t+1}) after a viewpoint selection and use it for the learning of the continuous VP policy. In the TRPO-based VP method (Liu et al., 2018b), the obtained viewpoint transitions will be discarded after they are trained due to the on-policy characteristic of TRPO. It leads to a low efficient use of the obtained transitions. In our work, the deterministic policy gradient theory (Silver et al., 2014) allows the agent to reuse the obtained transitions. Therefore, to make full use of the obtained viewpoint transitions, the experience replay (ER) (Lin, 1992; Schaul et al., 2016) technology is adopted and improved to implement a scheme of viewpoint transition management. The scheme includes viewpoint transition storage and viewpoint transition reuse.

3.4.1. Viewpoint transition storage

To store the obtained viewpoint transitions, we build a viewpoint transition buffer with a capacity of K in the light of Lin (1992) and Schaul et al. (2016). K is generally within $10^4 \sim 10^6$. Once the buffer is full of transitions, the old ones will be replaced by the newly generated transitions.

3.4.2. Viewpoint transition reuse

The key of viewpoint transition reuse is to decide which transitions to reuse. Lin (1992) adopt a uniform sampling strategy that means the sampling probability of each transition in the buffer is the same. However, those transitions with greater temporal difference (TD) errors are obviously more surprising to the agent and should be sampled with a higher probability (Schaul et al., 2016). Hence, Schaul et al. (2016) present a prioritized experience replay (PER) technology that can quantify the surprising level (priority) of each transition by the TD error and convert the priority into the corresponding sampling probability. Here, we employ the PER technology to sample the viewpoint transitions in the buffer. Concretely, the probability of sampling the ith stored viewpoint transition is given as

$$P(i) = \frac{p_i^\lambda}{\sum_{l=1}^{K} p_l^\lambda} \tag{7}$$

where $p_i^\lambda > 0$ is the priority of the ith transition. The exponent λ indicates how much prioritization is used, with $\lambda = 0$

corresponding to the uniform case. Proportional prioritization is defined with

$$p_i = |\hat{\delta}_i| + \epsilon \tag{8}$$

where $\hat{\delta}_i$ is the TD error of the ith transition and ϵ is a small positive value that prevents transitions with error of 0 from not being sampled. The estimation of TD error in PER is based on the double DQN algorithm (Mnih et al., 2015).

$$\hat{\delta}_i = r_t^{(i)} + \gamma Q(s_{t+1}^{(i)}, argmax_a Q(s_{t+1}^{(i)}, a; \omega); \omega^-) - Q(s_t^{(i)}, a_t^{(i)}; \omega) \tag{9}$$

where $Q(s_t, a_t; \omega)$ and $Q(s_t, a_t; \omega^-)$ are value function network and target value function network respectively. However, it is only applicable to discrete viewpoint planning, not to our continuous case. Inspired by Fujimoto et al. (2018), we improve the estimation method of TD error with the clipped double Q-learning algorithm so as to adapt to our deterministic continuous VP framework. The improved TD error is

$$\hat{\delta}_i = |\hat{y}_t^{(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_1)| + |\hat{y}_t^{(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_2)| \tag{10}$$

where $\hat{y}_t^{(i)} = r_t^{(i)} + \gamma \min_{j=1,2} Q(s_{t+1}^{(i)}, u(s_{t+1}^{(i)}; \theta^-); \omega_j^-)$ is TD target. $Q(s_t, a_t; \omega_1)$ and $Q(s_t, a_t; \omega_2)$ are two value function networks, and $Q(s_t, a_t; \omega_1^-)$ and $Q(s_t, a_t; \omega_2^-)$ are their corresponding target value function networks. $u(s_t; \theta^-)$ is the target policy network. These networks will be elaborated in the next subsection.

In addition, we find that the estimation of TD error is biased due to the prioritized sampling. It is known that Bellman optimality equation (Sutton and Barto, 2018) is $Q(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)}[r_t + \gamma \max_a Q(s_{t+1}, a)]$ where $y_t = r_t + \gamma \max_a Q(s_{t+1}, a)$ is TD target. Obviously, the distribution $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$ is changed by using the prioritized sampling, which introduces bias to the estimation of the expected value $Q(s_t, a_t)$. Thus, we correct the bias with importance-sampling weight $\rho = \frac{\mathcal{P}}{\mathcal{D}}$ where $\mathcal{D}$ is the new distribution of s_{t+1} generated due to the use of prioritized sampling. Then Bellman optimality equation is transformed to $Q(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathcal{D}(s_{t+1}|s_t, a_t)}[\rho(r_t + \gamma \max_a Q(s_{t+1}, a)]$ where $\rho(r_t + \gamma \max_a Q(s_{t+1}, a)$ is TD target with bias correction denoted as y_t^{corr}. And TD error is transformed to $\delta = y_t^{corr} - Q(s_t, a_t)$. Similar, in our scheme, the importance-sampling weight of the *ith* viewpoint transition in the buffer is

$$\rho_i = \frac{1}{K \cdot P(i)} \tag{11}$$

where K is the capacity of the buffer. Our clipped double Q-learning based TD error and TD target are corrected as

$$\hat{\delta}_i^{corr} = |\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_1)| + |\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_2)|$$
$$\hat{y}_t^{corr(i)} = \rho_i(r_t^{(i)} + \gamma \min_{j=1,2} Q(s_{t+1}^{(i)}, u(s_{t+1}^{(i)}; \theta^-); \omega_j^-)). \tag{12}$$

To avoid expensive sweeps over the entire viewpoint transition buffer, priorities are only updated for the transitions that are sampled according to Schaul et al. (2016). In addition, the new transitions will be put in the buffer with maximal priority in order to guarantee that all transitions are seen at least once.

```
Input: Parameters: $\sigma_1, N, \sigma_2, c, \beta, d, \alpha, \tau, K$
Output: $\theta$
Initialize the value function networks
  $Q(s_t, a_t; \omega_1), Q(s_t, a_t; \omega_2)$, and the VP policy network
  $u(s_t; \theta)$ with random parameters $\omega_1, \omega_2, \theta$
Initialize the target networks
  $\omega_1^- \leftarrow \omega_1, \omega_2^- \leftarrow \omega_2, \theta^- \leftarrow \theta$
Initialize the viewpoint transition buffer $\mathcal{B}$
  with the capacity $K$
for $t = 1$ to $T$ do
    Run a behavioral policy with exploration noise
      to select an action $\tilde{a}_t \sim u(s_t; \theta) + \epsilon_1, \epsilon_1 \sim \mathcal{N}(0, \sigma_1)$
      and receive a reward $r_t$ and a new state $s_{t+1}$
    Store the transition tuple $(s_t, \tilde{a}_t, r_t, s_{t+1})$ in $\mathcal{B}$
      with maximal priority
    for $i = 1$ to $N$ do
        Sample transitions $(s_t^{(i)}, \tilde{a}_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)})$ from the
          buffer $\mathcal{B}$: $i \sim P(i) = \frac{p_i^\lambda}{\sum_{i=1}^K p_i^\lambda}$ (Equation 7)
        Compute importance-sampling weight $\rho_i$
          (Equation 11)
        Estimate the corrected TD targets $\hat{y}_t^{corr(i)}$
          using Equation (12)
        Compute $\tilde{a}_{t+1} = u(s_{t+1}; \theta^-) + \epsilon_2, \epsilon_2 \sim clip(\mathcal{N}(0, \sigma_2), -c, c)$
          according to the smoothing regularization
          of TD3 (Fujimoto et al., 2018)
        Estimate the corrected TD error $\hat{\delta}_i^{corr}$
          (Equation 12)
        Update transition priority using Equation
          (8)
    Update the value function networks by
      optimizing the objective (Equation 14):
      $\omega_j = \omega_j - \beta \nabla_{\omega_j} J(\omega_j)$
    if $t \% d == 0$ then
        Update the policy network using the gradient
          (Equation 13):
          $\theta = \theta + \alpha \frac{1}{N} \sum_{i=1}^N [\rho_i \cdot \nabla_\theta u(s_t^{(i)}; \theta) \nabla_a Q(s_t^{(i)}, u(s_t^{(i)}; \theta); \omega_1)]$
        Update the target networks:
        $\omega_j^- = \tau \omega_j + (1 - \tau) \omega_j^-$
        $\theta^- = \tau \theta + (1 - \tau) \theta^-$
return $\theta$
```

Algorithm 1. Training the deterministic continuous VP policy network.

3.5. Training the policy network

In this section, we resort twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the scheme designed in Section 3.4 to develop a training algorithm for the solution of the optimal VP policy parameters θ^*. To this end, we use the gradient (Equation 4) to iteratively update θ: $\theta = \theta + \alpha \nabla_\theta J(\theta)$. α is the learning rate. The core task is to solve the gradient $\nabla_\theta J(\theta)$. We therefore employ Monte Carlo method to replace the expected operator in Equation (4) in an approximate manner. Specifically, we sample N transitions from the viewpoint transition buffer using Equation (7) to calculate

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} [\rho_i \cdot \nabla_\theta u(s_t^{(i)}; \theta) \nabla_a Q^u(s_t^{(i)}, u(s_t^{(i)}; \theta))]. \tag{13}$$

According to TD3, we approximately represent the value function $Q^u(s_t, a_t)$ in Equation (13) by a three-layer fully-connected neural network $Q(s_t, a_t; \omega)$ with the parameters ω. The network takes the state s_t and the action a_t as input and outputs the function value $Q(s_t, a_t; \omega)$. By updating the parameters ω, the value function corresponding to the VP policy u can be obtained.

In order to better train the policy network $u(s_t; \theta)$, we follow TD3 to build six neural networks in total: policy network $u(s_t; \theta)$, value function network 1 $Q(s_t, a_t; \omega_1)$, value function network 2 $Q(s_t, a_t; \omega_2)$ and their corresponding target networks [target policy network $u(s_t; \theta^-)$, target value function network 1 $Q(s_t, a_t; \omega_1^-)$, target value function network 2 $Q(s_t, a_t; \omega_2^-)$]. After the training, the policy network $u(s_t; \theta)$ is the optimal deterministic continuous VP policy we want. The other networks only serve as auxiliary training. Figure 3 shows the relationship between the six networks.

The value function networks can be updated with the aforementioned N samples by minimizing the objective

$$L(\omega_j) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_j))^2 \tag{14}$$

where j is 1 or 2. $\hat{y}_t^{corr}$ is the corrected TD target proposed in Equation (12).

Our whole algorithm to train the deterministic continuous VP policy network is summarized in Algorithm 1. Once the optimal parameters θ^* are obtained after the training, we can use them for the practical AOR task. Given a state s_t, the planned action is $a_t^* = u(s_t; \theta^*)$, and the next best viewpoint of the agent is $\Phi_{t+1} = \Phi_t + a_t^*$.

4. Experiments

This section first provides details about the experimental dataset and implementation, and then reports the experimental results along with some analyzes.

4.1. Dataset and metric

We evaluate our proposed deterministic continuous VP method on the public and challenging dataset GERMS (Malmir et al., 2015) shown in Figure 4A which is collected in the context of developing robots to interact with toddlers in early childhood education environments. The dataset has 1,365 video tracks of give-and-take trials using 136 different object instances. The object instances are soft toys denoting a wide range of disease-related organisms, microbes and human cell types. Each video track records a robot grasping an object instance to its center of view, rotating the object by 180° with its left or right arm, and then returning it. All video tracks were recorded by a head-mounted camera of the robot at 30 frames/s, as shown in Figure 4B. At the same time, the joint position and object label corresponding to each frame image were also recorded in each track. These joint positions provide an opportunity for verifying different VP methods in one dimensional action space. The dataset authors specified the image subsets of all tracks as train and test set, as shown in Table 1. The evaluation metric used for different VP methods is recognition accuracy that is the average value of the entire test set. The higher the recognition accuracy is, the better the corresponding VP method will be.

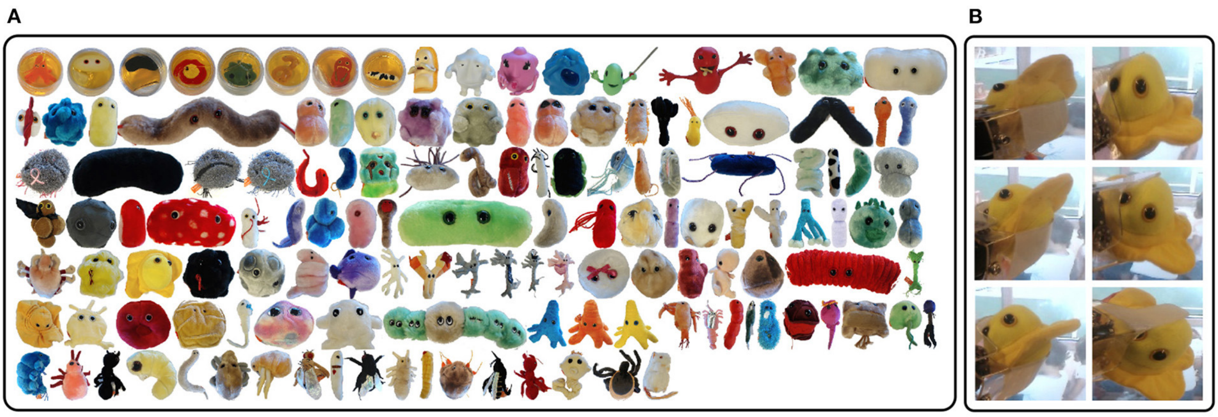


FIGURE 4
The GERMS dataset. **(A)** One hundred and thirty six object instances. **(B)** Recorded images of different joint positions in each track.

4.2. Implementation details

4.2.1. Network architecture

The Tensorflow platform is used to implement the proposed method in this work. In the pre-trained classifier, we transform every image in the GERMS dataset into a 4,096-dimensional feature vector using an existing CNN model VGG-net provided by Malmir et al. (2015). The *softmax* layer has 136 neurons. For the policy network $u(s_t;\theta)$, the dimensions of each layer are 136, 512, 512 and 1. The activation functions of the two hidden layers are both *relu*. The output layer adopts *tanh* activation function, which is multiplied by 512 so as to make the planned relative VP action in $[-45^\circ, 45^\circ]$. For the two value function networks ($Q(s_t, a_t;\omega_1)$ and $Q(s_t, a_t;\omega_2)$), they have the same network structure with the dimensions of each layer are 137, 512, 512 and 1. The activation functions of the two hidden layers are also *relu*. The configuration of their corresponding target network is completely consistent with theirs.

4.2.2. Viewpoint transition management

The capacity of the viewpoint transition buffer is 10^6. ϵ and the exponent λ are set as 0.01 and 0.6 according to the original setting of PER (Schaul et al., 2016). To efficiently sample from distribution (Equation 7), we use a "sum-tree" (Schaul et al., 2016) in which every node is the sum of its children and the leaf nodes are priorities. The sum-tree can be efficiently updated and sampled from.

TABLE 1 GERMS dataset statistics (mean ± std).

Images/track	Number of tracks	Images/track	Total number of images
Train	816	157 ± 12	76,722
Test	549	145 ± 19	51,561

4.2.3. Training

The reward discount factor γ is 0.95. The minibatch size N is 128. The maximum step T_{max} for recognition is $T_{max} = 12$ and the preset probability threshold is 0.99. The Adam optimizer (Kingma and Ba, 2014) is utilized to optimize the policy network and the value function networks. The learning rates are 0.0001, 0.001, and 0.001, respectively. The standard deviations (σ_1 and σ_2) of the exploration noise and smoothing regularization are 128 and 32. c is 512. The delayed update cycle d and soft update τ are 2 and 0.01.

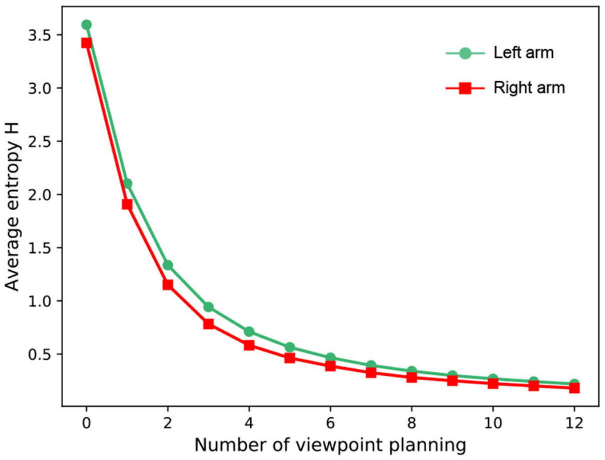


FIGURE 6
The average entropy over the whole test dataset. The experiment is implemented with our VP model.

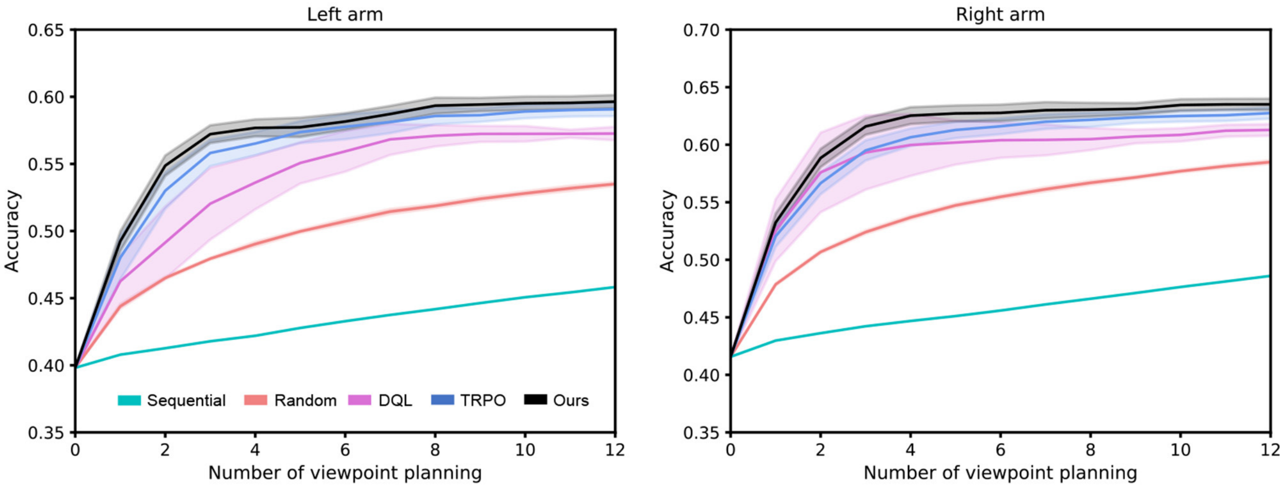


FIGURE 5
Performance comparison between our presented deterministic continuous VP approach and several competing methods. The shaded region represents the standard deviation of the average evaluation over 10 trials.

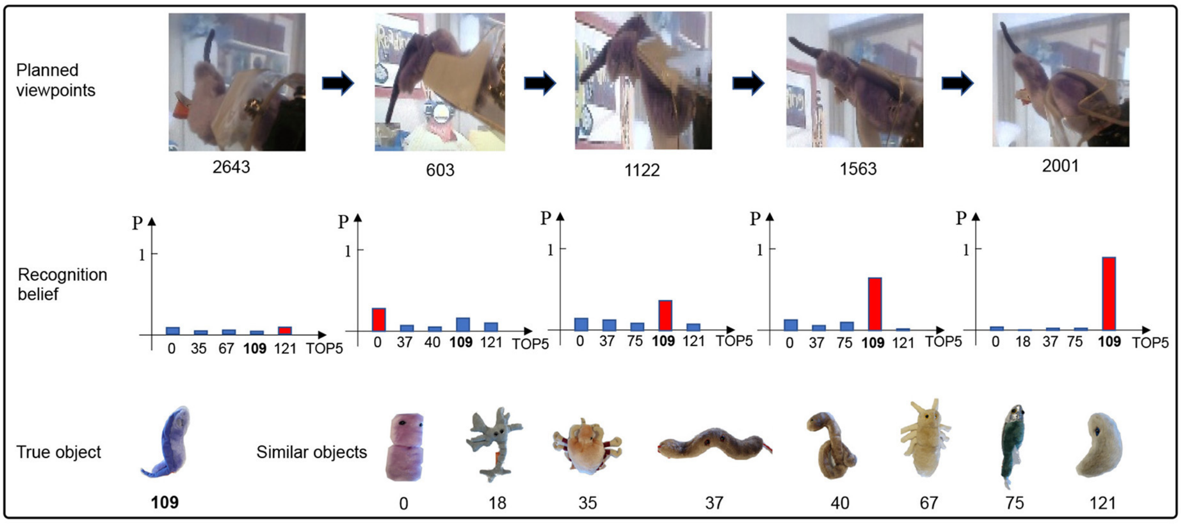


FIGURE 7
An example of actively identifying an object by our VP method. The recognition belief increases with the increase of the number of viewpoint planning.

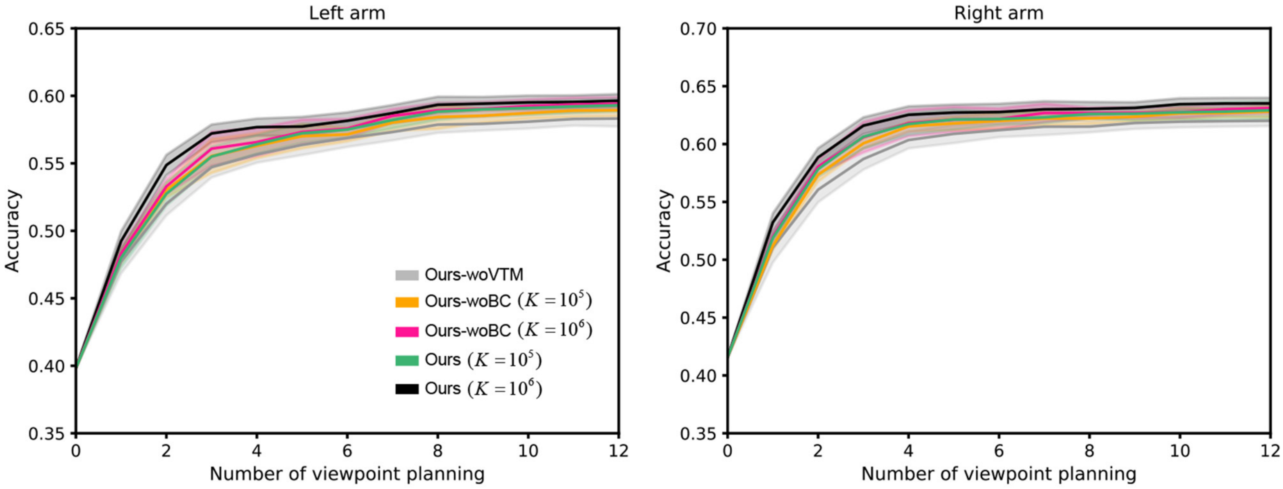


FIGURE 8
The performance comparison results of ablation experiments. *K* represents the capacity of the viewpoint transition buffer. The shaded region represents the standard deviation of the average evaluation over 10 trials.

4.3. Results and analyzes

4.3.1. Comparison with competing methods

To validate the effectiveness of our proposed deterministic continuous VP method in this experiment, we compare our proposed method with the following baseline and competing methods.

4.3.1.1. Single viewpoint recognition

Single viewpoint recognition only allows the agent to recognize an object from one viewpoint.

4.3.1.2. Blind VP policies

Random policy (Liu et al., 2018a) randomly selects an action from the continuous action space $[-45^\circ, 45^\circ]$ with a uniform probability. Sequential policy (Liu et al., 2018a) moves the agent to the next adjacent viewpoint in the same direction. The reason why these two baseline policies are called blind VP policies is that they do not use the previous observation information for purposeful viewpoint planning. The blind policies may produce worthless viewpoints for recognition.

4.3.1.3. Purposeful discrete VP policy

DQL policy (Malmir et al., 2015; Malmir and Cottrell, 2017) develops an active discrete VP method with deep Q-Learning algorithm, which explores in the discrete action space $\{\pm\frac{\pi}{64}, \pm\frac{\pi}{32}, \pm\frac{\pi}{16}, \pm\frac{\pi}{8}, \pm\frac{\pi}{4}\}$.

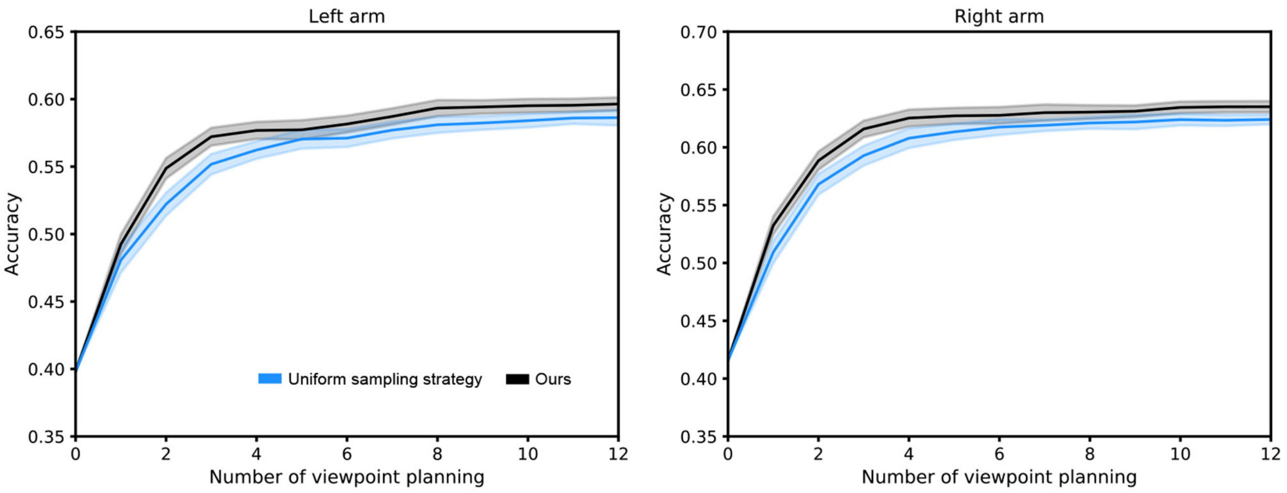


FIGURE 9
Performance comparison between our sampling strategy and uniform sampling strategy. The capacity of the viewpoint transition buffer is 10^6. The shaded region represents the standard deviation of the average evaluation over 10 trials.

4.3.1.4. Purposeful continuous VP policy

TRPO policy (Liu et al., 2018b) utilizes trust region policy optimization (Schulman et al., 2015) to learn a continuous VP policy and adopts extreme learning machine (Huang et al., 2006) to reduce computational complexity. This policy has on-policy characteristic that means the agent can not reuse learned viewpoint transitions for efficient training.

Since the main focus of this work is viewpoint planning, we do not investigate the impact of classifiers on recognition performance. Therefore, for a fair comparison, the classifiers in different approaches are the same in the experiment. Figure 5 reports the experimental results of our method against other approaches over 10 random seeds of the policy network initialization. Some observations from Figure 5 are presented as follows: (1) Viewpoint planning can greatly improve recognition performance. The number of VP is 0 that means the agent recognizes the concerned object with a single viewpoint. Obviously, the recognition accuracy of single viewpoint recognition policy is far lower than that of the methods which perform multi viewpoint recognition *via* VP. This is because more object information with difference can be found through VP to reduce recognition uncertainty, thus improving the recognition performance. As shown in Figure 6, the uncertainty of recognition decreases as the number of viewpoints increases. Figure 7 shows the process of actively identifying an object. (2) The performance of the blind VP policies is nowhere near as good as that of the purposeful VP policies. The primary reason is that the purposeful VP policies (i.e., DQL policy, TRPO policy and our policy) can purposefully plan next viewpoints according to the observed information. (3) The continuous VP policies have better performance than the discrete VP policy. That is because the continuous VP policies (i.e., TRPO policy and our policy) directly explore continuous viewpoint space without sampling, so they will not miss some important viewpoints. (4) The performance of our deterministic continuous VP policy exceeds that of TRPO policy. This is mainly because we design a scheme of viewpoint transition management that can reuse the obtained viewpoint transitions to improve the training effect.

4.3.2. Ablation studies

To verify the importance of different components in our proposed VP model, we intend to conduct the variant experiments with the ablation of different components, i.e., viewpoint transition management (VTM) and bias correction (BC). Training the model without VTM and BC are respectively denoted as Ours-woVTM and Ours-woBC. From the presented results over 10 random seeds in Figure 8, we can notice that: (1) The performance of Ours-woVTM is the worst. It illustrates that our designed scheme of viewpoint transition management indeed enhances the training effect. (2) The performance of Ours-woBC is inferior to that of Ours, especially when the capacity K of the viewpoint transition buffer is large. This is because when the capacity is larger, the distribution of s_{t+1} in the buffer is closer to its true distribution. In this case, the effect of our bias correction based on importance sampling will be more obvious.

4.3.3. Sampling strategies investigations

To verify the superiority of our proposed sampling strategy (i.e., prioritized experience replay based on clipped double Q-learning and bias correction) in the scheme of viewpoint transition management, we conduct comparison experiments with the uniform sampling strategy (Lin, 1992) over 10 random seeds. As shown in Figure 9, we observe that our sampling strategy achieves a better performance, since the importance of each viewpoint transition is ignored by the uniform sampling strategy.

5. Conclusions

In this paper, a continuous viewpoint planning method with transition management is proposed for active object recognition based on reinforcement learning. Specifically, we employ deterministic policy gradient theory to build a learning framework of the viewpoint planning policy. We also design a scheme of viewpoint transition management that can store and reuse the obtained transitions. We develop an algorithm based on twin delayed deep deterministic gradient and the designed scheme to train the policy. Experiments on a public dataset demonstrate the effectiveness of our method. In the future, we will integrate the calibrated probabilistic classifiers in AOR research. As stated in Popordanoska et al. (2022), the way the posterior probability distribution is defined in our work assumes that the classifier is properly calibrated, i.e. the *softmax* output represents the correct error rate probabilities. In general, this is not necessarily the case.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

HS and YL: conceptualization. FZ and HS: methodology. HS and YK: software. HS and SF: investigation. FZ: resources and funding acquisition. YL: data curation. HS: writing—original draft. YL and PZ: writing—review and editing. JW and YW: supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the National Natural Science Foundation of China under Grant No. U1713216.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Andreopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: directions forward. *Comput. Vis. Image Understand.* 117, 827–891. doi: 10.1016/j.cviu.2013.04.005

Becerra, I., Valentin-Coronado, L. M., Murrieta-Cid, R., and Latombe, J.-C. (2014). "Appearance-based motion strategies for object detection," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 6455–6461.

Behl, A., Hosseini Jafari, O., Karthik Mustikovela, S., Abu Alhaija, H., Rother, C., and Geiger, A. (2017). "Bounding boxes, segmentations and object coordinates: how important is recognition for 3d scene flow estimation in autonomous driving scenarios," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2574–2583.

Bellemare, M. G., Dabney, W., and Munos, R. (2017). "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning* (PMLR), 449–458. doi: 10.48550/arXiv.1707.06887

Duan, J., Bello, G., Schlemper, J., Bai, W., Dawes, T., Biffi, C., et al. (2019). Automatic 3D bi-ventricular segmentation of cardiac images by a shape-refined multi-task deep learning approach. *IEEE Trans.* 38, 2151–2164. doi: 10.1109/TMI.2019.2894322

Fujimoto, S., Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning* (PMLR), 1587–1596. doi: 10.48550/arXiv.1802.09477

Hao, G., Fu, Z., Feng, X., Gong, Z., Chen, P., Wang, D., et al. (2021). A deep deterministic policy gradient approach for vehicle speed tracking control with a robotic driver. *IEEE Trans. Autom. Sci. Eng.* 19, 2514–2525. doi: 10.1109/TASE.2021.3088004

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing* 70, 489–501. doi: 10.1016/j.neucom.2005.12.126

Jayaraman, D., and Grauman, K. (2019). End-to-end policy learning for active visual categorization. *IEEE T. Pattern Anal.* 41, 1601–1614. doi: 10.1109/TPAMI.2018.2840991

Kingma, D. P., and Ba, J. (2014). ADAM: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* doi: 10.48550/arXiv.1412.6980

Liang, Y., Guo, C., Ding, Z., and Hua, H. (2020). Agent-based modeling in electricity market using deep deterministic policy gradient algorithm. *IEEE Trans. Power Syst.* 35, 4180–4192. doi: 10.1109/TPWRS.2020.2999536

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971.*

Lin, L.-J. (1992). *Reinforcement Learning for Robots Using Neural Networks.* Pittsburgh, PA: Carnegie Mellon University.

Liu, H., Li, F., Xu, X., and Sun, F. (2018a). Active object recognition using hierarchical local-receptive-field-based extreme learning machine. *Memet. Comput.* 10, 233–241. doi: 10.1007/s12293-017-0229-2

Liu, H., Wu, Y., and Sun, F. (2018b). Extreme trust region policy optimization for active object recognition. *IEEE Trans. Neural Network Learn. Syst.* 29, 2253–2258. doi: 10.1109/TNNLS.2017.2785233

Malmir, M., and Cottrell, G. W. (2017). "Belief tree search for active object recognition," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), 4276–4283.

Malmir, M., Sikka, K., Forster, D., Movellan, J. R., and Cottrell, G. (2015). "Deep q-learning for active recognition of germs: Baseline performance on a standardized dataset for active learning," in *Proceedings of the British Machine Vision Conference (BMVC)*, 161.1–161.11. doi: 10.5244/C.29.161

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. *arXiv preprint.* doi: 10.48550/arXiv.1312.5602

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Paletta, L., and Pinz, A. (2000). Active object recognition by view integration and reinforcement learning. *Robot. Auton. Syst.* 31, 71–86. doi: 10.1016/S0921-8890(99)00079-2

Parr, T., Sajid, N., Da Costa, L., Mirza, M. B., and Friston, K. J. (2021). Generative models for active vision. *Front. Neurorobot.* 15, 651432. doi: 10.3389/fnbot.2021.651432

Patten, T., Zillich, M., Fitch, R., Vincze, M., and Sukkarieh, S. (2015). Viewpoint evaluation for online 3-d active object classification. *IEEE Robot. Autom. Lett.* 1, 73–81. doi: 10.1109/LRA.2015.2506901

Popordanoska, T., Sayer, R., and Blaschko, M. B. (2022). A consistent and differentiable lp canonical calibration error estimator. *arXiv preprint*. doi: 10.48550/arXiv.2210.07810

Potthast, C., Breitenmoser, A., Sha, F., and Sukhatme, G. S. (2016). Active multi-view object recognition: a unifying view on online feature selection and view planning. *Robot Auton. Syst.* 84, 31–47. doi: 10.1016/j.robot.2016.06.013

Roynard, X., Deschaud, J.-E., and Goulette, F. (2018). "Paris-lille-3d: a point cloud dataset for urban scene segmentation and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR)* (Salt Lake City, UT: IEEE), 2027–2030.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). "Prioritized experience replay," in *Proceedings of the International Conference on Learning Representations (ICLR)*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). "Trust region policy optimization," in *International Conference on Machine Learning* (PMLR), 1889–1897. doi: 10.48550/arXiv.1502.05477

Shi, Q., Lam, H.-K., Xuan, C., and Chen, M. (2020). Adaptive neuro-fuzzy pid controller based on twin delayed deep deterministic policy gradient algorithm. *Neurocomputing* 402, 183–194. doi: 10.1016/j.neucom.2020.03.063

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). "Deterministic policy gradient algorithms," in *International Conference on Machine Learning* (PMLR), 387–395.

Stria, J., and Hlavác, V. (2018). "Classification of hanging garments using learned features extracted from 3d point clouds," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 5307–5312.

Sun, Y., Ran, X., Zhang, G., Wang, X., and Xu, H. (2020). Auv path following controlled by modified deep deterministic policy gradient. *Ocean Eng.* 210, 107360. doi: 10.1016/j.oceaneng.2020.107360

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Van de Maele, T., Verbelen, T., Çatal, O., and Dhoedt, B. (2022). Embodied object representation learning and recognition. *Front. Neurorobot.* 16, 840658. doi: 10.3389/fnbot.2022.840658

Wang, Y., Sun, J., He, H., and Sun, C. (2020). Deterministic policy gradient with integral compensator for robust quadrotor control. *IEEE Trans. Syst. Man Cybernet. Syst.* 50, 3713–3725. doi: 10.1109/TSMC.2018.2884725

Wu, J., Yang, Z., Liao, L., He, N., Wang, Z., and Wang, C. (2022). A state-compensated deep deterministic policy gradient algorithm for uav trajectory tracking. *Machines* 10, 496. doi: 10.3390/machines10070496

Wu, K., Ranasinghe, R., and Dissanayake, G. (2015). "Active recognition and pose estimation of household objects in clutter," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 4230–4237.

Zeng, R., Wen, Y., Zhao, W., and Liu, Y.-J. (2020). View planning in robot active vision: a survey of systems, algorithms, and applications. *Comput. Vis. Media* 6, 225–245. doi: 10.1007/s41095-020-0179-3

Zhang, T., Zhu, K., and Wang, J. (2020). Energy-efficient mode selection and resource allocation for d2d-enabled heterogeneous networks: a deep reinforcement learning approach. *IEEE T. Wirel. Commun.* 20, 1175–1187. doi: 10.1109/TWC.2020.3031436

Zhao, D., Chen, Y., and Lv, L. (2016). Deep reinforcement learning with visual attention for vehicle classification. *IEEE Tran. Cogn. Dev. Syst.* 9, 356–367. doi: 10.1109/TCDS.2016.2614675

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 06 July 2023
DOI 10.3389/fnbot.2023.1039644

Check for updates

OPEN ACCESS

EDITED BY
Yimin Zhou,
Chinese Academy of Sciences (CAS), China

REVIEWED BY
A. B. Ferozkhan,
C. Abdul Hakeem College of Engineering and Technology, India
Eiji Uchibe,
Advanced Telecommunications Research Institute International (ATR), Japan

*CORRESPONDENCE
Zhongxue Gan
✉ ganzhongxue@fudan.edu.cn
Yi Liu
✉ liuyi_@fudan.edu.cn

RECEIVED 08 September 2022
ACCEPTED 19 June 2023
PUBLISHED 06 July 2023

CITATION
Li W, Liu Y, Ma Y, Xu K, Qiu J and Gan Z (2023) A self-learning Monte Carlo tree search algorithm for robot path planning.
Front. Neurorobot. 17:1039644.
doi: 10.3389/fnbot.2023.1039644

A self-learning Monte Carlo tree search algorithm for robot path planning

Wei Li[1], Yi Liu[1]*, Yan Ma[1], Kang Xu[1], Jiang Qiu[1] and Zhongxue Gan[1,2]*

[1]Academy for Engineering and Technology, Fudan University, Shanghai, China, [2]Ji Hua Laboratory, Department of Engineering Research Center for Intelligent Robotics, Foshan, China

This paper proposes a self-learning Monte Carlo tree search algorithm (SL-MCTS), which has the ability to continuously improve its problem-solving ability in single-player scenarios. SL-MCTS combines the MCTS algorithm with a two-branch neural network (PV-Network). The MCTS architecture can balance the search for exploration and exploitation. PV-Network replaces the rollout process of MCTS and predicts the promising search direction and the value of nodes, which increases the MCTS convergence speed and search efficiency. The paper proposes an effective method to assess the trajectory of the current model during the self-learning process by comparing the performance of the current model with that of its best-performing historical model. Additionally, this method can encourage SL-MCTS to generate optimal solutions during the self-learning process. We evaluate the performance of SL-MCTS on the robot path planning scenario. The experimental results show that the performance of SL-MCTS is far superior to the traditional MCTS and single-player MCTS algorithms in terms of path quality and time consumption, especially its time consumption is half less than that of the traditional MCTS algorithms. SL-MCTS also performs comparably to other iterative-based search algorithms designed specifically for path planning tasks.

KEYWORDS

Monte Carlo tree search (MCTS), path planning, neural network, Markov decision process (MDP), collective intelligent algorithm

1. Introduction

Path planning is a critical problem in logistics and robotics and has been further applied to many areas (Zhang et al., 2019; Aggarwal and Kumar, 2020; Li et al., 2021). The objective of path planning is to obtain an optimal and collision-free path from the origin to the destination. In recent years, collective intelligence algorithms have been widely used for path planning. These algorithms solve path planning problems by simulating some natural phenomenon or biological behaviors such as particle swarm optimization (Cheng et al., 2021; Halder, 2021; Yu et al., 2022), ant colony optimization (ACO) (Xiong et al., 2021), and genetic algorithm (Lee and Kim, 2016). The collective intelligence algorithm is based on the iterative search to find the solution but typically suffers from poor solution quality, slow convergence and inefficient search (Dai et al., 2019; Cheng et al., 2021).

Monte Carlo tree search (MCTS) is an iterative approach which executes random sampling in the simulation and collects action statistics to enable educated choice in subsequent iterations. Since the number of simulations in each iteration can be considered the number of agents searching in the state space, it is also regarded as a collective intelligence algorithm (Qi et al., 2018, 2021). Agents find a reasonable solution, and then refine it to find an optimal one in the subsequent iteration. One of the most significant advantages of MCTS is that the algorithm does not require domain-specific knowledge, with only search rules specifying which actions are possible and which are terminated in each state. It allows MCTS to be used in any task that can be modeled with decision trees (although it may be helpful to add domain-specific knowledge). Moreover, MCTS can run additional iterations to improve its performance. In particular, MCTS is biased toward more promising states when adding nodes to the search tree. These properties of MCTS make its search process faster than most collective intelligence algorithms. However, with the increasing number of simulations, its search speed also becomes slow. This work proposes an algorithmic framework of self-learning MCTS to address this problem.

MCTS is often adopted in applications, such as games (Crippa et al., 2022), combinatorial optimization problems (Perez et al., 2012), planning problems (Pellier et al., 2010; Dam et al., 2022), and scheduling problems (Huang et al., 2022; Kung et al., 2022). MCTS was initially proposed by Gelly and Wang (2006). Later, Kocsis and Szepesvári (2006) developed MCTS as the first computer Go program, and MCTS rapidly gained widespread attention due to its significant success in playing Go. While some new work applies MCTS and its variations on tasks such as two-player games (Gelly et al., 2012) and multi-player games (Sturtevant, 2008; Scariot et al., 2022), so far there is only a little work about single-player tasks (Schadd et al., 2012). For SameGame, Schadd et al. (2012) proposed Single Player Monte Carlo Tree Search (SP-MCTS) to improve the performance of MCTS on this single-player game. SP-MCTS overperformed previous works in single-player deterministic complete information games by adjusting the selection and back-propagation strategies. Furthermore, Crippa et al. (2022) improved the performance of SP-MCTS in SameGame by solving the deadlock problems. Dam et al. (2022) tried to use MCTS to find feasible solutions in robot path planning. This work shows that a suitable sampling range, hyper-parameter of sampling configuration and exploration strategies could substantially boost the performance of MCTS significantly. In summary, the MCTS algorithms mentioned above are based on the conventional MCTS framework, i.e., they focus on solving a single problem through a large number of random searches in the simulation process, which is a greedy way to find a solution. It leads the search process to be inefficient.

In recent years, the outstanding performance of AlphaGo Zero in playing the game Go (Silver et al., 2016, 2017) further highlighted the capabilities of MCTS. The critical characteristic of AlphaGo Zero is to assess each game's trajectories based on the self-play results. However, self-play in two-player zero-sum scenarios is based on game relationships, and it is not directly transferable to be used in single-player scenarios. The main challenge is evaluating the current model's solution quality in the environment without the game relationship. In this paper, we construct a self-learning approach for single-player tasks, which enables the single-player MCTS to improve its problem-solving ability by learning from its historical experience. The proposed self-learning MCTS (SL-MCTS) combines MCTS with a neural network (PV-Network). The framework of MCTS can balance the exploration and exploitation of search. PV-Network replaces the rollout process of the traditional MCTS framework and predicts the search probability of each subsequent move and the state value, which reduces the operational time of SL-MCTS. This work presents a method to evaluate the performance of the current model's solution for the self-learning process of SL-MCTS by comparing the current model's performance with the solution obtained from the best historical model so far. The current solution is scored higher (lower) if better (worse) than the previous optimal solution. This method can guide PV-Network to make predictions accurately, increasing the effectiveness of SL-MCTS search. SL-MCTS generates training data based on the solutions of the current model and their corresponding scores. In the self-learning process, PV-Network improves its selection probability and score prediction accuracy by learning the historical experience of SL-MCTS. The enhanced PV-Network can, in turn, guide SL-MCTS to find a better solution. The above process is repeated to gradually improve the problem-solving ability of SL-MCTS. In this paper, we validated the effectiveness of the proposed method in the classic and widely used path planning scenario.

The main contributions of this paper are summarized as follows:

1. We propose a self-learning framework to continuously improve the problem-solving ability of SL-MCTS in a single-player environment.
2. This study proposes a method to evaluate decision quality in single-player scenarios, which utilizes the best historical models. By utilizing this evaluation method, the SL-MCTS algorithm can consistently and effectively enhance its decision-making capacity in single-player scenarios.
3. We demonstrate that SL-MCTS effectively improves problem-solving ability through self-learning process in robot path planning scenario. Comparisons with other MCTS algorithms and collective intelligence algorithms also confirm the superior efficiency of SL-MCTS.

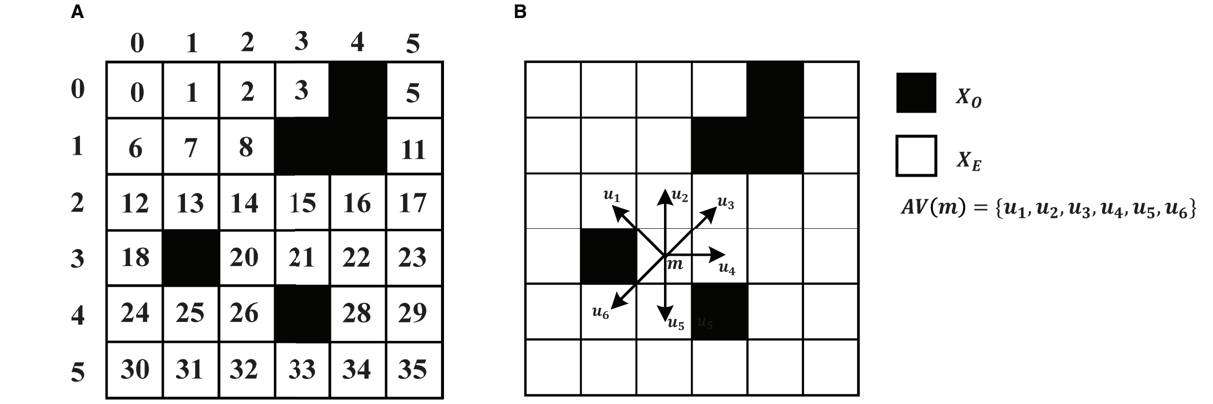
The rest of this paper is organized as the following. Section 2 presents the construction of environmental maps and the definition of the path planning problem in this paper, the procedure of conventional MCTS algorithms, and the detail of SL-MCTS algorithm. Section 3 provides the experimental setting and experimental results of SL-MCTS. We also compare the performance of SL-MCTS with traditional MCTS SP-MCTS algorithms and other collective intelligence algorithms in robot path planning scenarios. The paper is concluded in Section 4, where we also discuss ideas for future works.

2. Materials and methods

2.1. Problem formulation

2.1.1. Path planning problem

This paper utilizes the grid model to form the robot's working environment for path planning tasks. As shown in Figure 1A, the space is partitioned into $N \times N$ blocks, whereby the black grids represent obstacles (grids with barriers), and the white grids represent free space (areas where the robot can move). To identify obstacles, white grid cells are represented by 0, whereas back grid units are represented by 1.



	0	1	2	3	4	5
0	0	1	2	3	■	5
1	6	7	8	■	■	11
2	12	13	14	15	16	17
3	18	■	20	21	22	23
4	24	25	26	■	28	29
5	30	31	32	33	34	35

FIGURE 1
(A) Environment model. **(B)** An example of a two-dimensional path planning problem with eight directions.

Figure 1B is an example of a 6×6 grid map. The task information includes a pair of origins and destinations. The set of all nodes is denoted as X, where X_O denotes the set of obstacles and X_E contains all the feasible. The origin and destination are respectively denoted as m_s and m_d. The relationship between all feasible nodes (X_E) is denoted as $G = (M, E)$, where $M \in X_E$ and E is the edges to neighbor nodes of M. $AE(m) = \{m' \mid (m, m') \in E, m \neq m'\}$ represents all feasible neighbors of node m. $N(AE(m))$ is the number of the feasible nodes of m. The cost of each edge is recorded as 1. Therefore the path planning can be described as an agent starting from position x_s at time step t_0 to position x_d. At time step t_1, the agent selects action a_1 and moves to the next state s_1. After T steps, the agent reaches the position x_d in S_T. The sequential solution is $path = ((a_0, a_1, \ldots, a_t, \ldots, a_T), a_0 = x_s, a_T = x_d, a_t \in AE(a_{t-1}))$, and the path length is $\sum_{t=0}^{t=T-1} cost(a_t, a_{t+1})$.

2.1.2. Markov decision process of path planning

We model the search process of path planning as a Markov Decision Process (MDP). The process can be described as shown in Figure 2. At each time step, the map is defined as state $S_t(t = 0, 1, 2, 3, ..., T)$. The neural network predicts the state value v_t and the selection probabilities $\boldsymbol{p_t}$ for each state S_t. The choice of action a_{t+1} is together determined by v_t and $\boldsymbol{p_t}$, executing action a_{t+1} and transferring to the next state S_{t+1}. This process continues until the agent reaches the end.

2.2. Monte Carlo tree search

To explicitly compare the differences between the framework of traditional MCTS and that of SL-MCTS, we describe the flow of the traditional MCTS algorithm in this part. For the family of traditional MCTS algorithms, their steps are similar.

Two fundamental concepts guide the search process of MCTS algorithms: (1) the true value of an action can be approached by a large number of stochastic simulations; (2) these values can be effectively used to adjust the policy to the best-preferred strategy. MCTS builds a search tree to estimate the values of the moves. These estimates (especially those of the most promising directions) become more and more accurate as the iterative search increases. Generally, the basic MCTS algorithm has four main processes (as shown in Figure 3): selection, expansion, simulation, and backpropagation. The tree policy is used to balance exploration and exploitation in the search and also determines the search direction. The Default Policy aims to calculate the action value of the non-terminal state by rapidly exploring a certain depth of the tree in the rollout. The rollout subtree provides the statistics for MCTS decision-making. The general approach of the rollout is to select actions based on uniform distribution. In the rollout process, a quick search is performed according to the default policy to produce a rollout subtree and find a result until the limits on the maximum number of iterations and the maximum depth of exploration are reached. In general, with a larger number of search depths and iterations, MCTS performs well, but it also causes the problem of inefficient search.

2.3. SL-MCTS algorithm

Algorithm 1 presents the pseudocode of SL-MCTS. SL-MCTS combines MCTS with a two-branch neural network (PV-Network) which guides the evaluation phase of SL-MCTS (Figure 4). The search process of SL-MCTS is shown in Algorithm 1, lines 3-12. PV-Network has two branches that output the selection probabilities $\boldsymbol{p}$ of all feasible nodes and a state value v, respectively (line 7). The selection probability $\boldsymbol{p}$ of each node is output after the search (line 10). At the end of the task, the solution score z is evaluated by comparing it with the optimal historical model (line 13), which means the quality of paths. The training process is shown in lines 15–20. In the training process, the parameters of PV-Network are updated, which makes the select probability $\boldsymbol{p}$ and state value v closer to the search probability $\boldsymbol{\pi}$ and path quality score z of previous SL-MCTS (line 16). Finally, these new network parameters are used in the next iteration of self-learning to make the search direction of SL-MCTS more accurate. The map and historical path information are fused as input state S_t. The selection probability $\boldsymbol{p}$ is

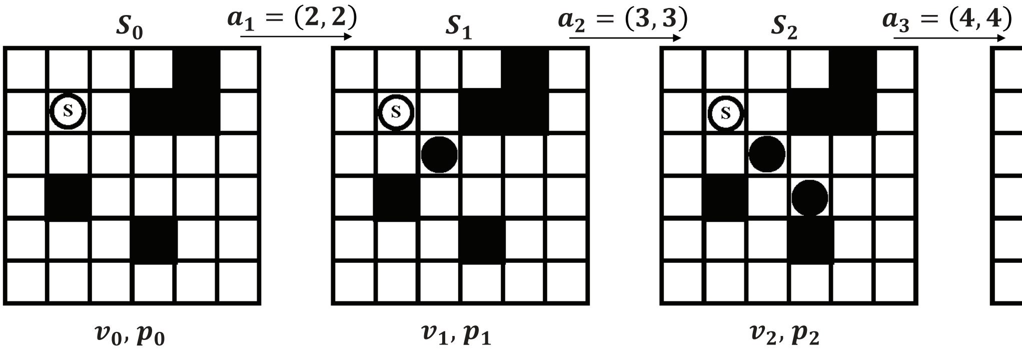


FIGURE 2
Modeling path planning problem as a Markov decision process.

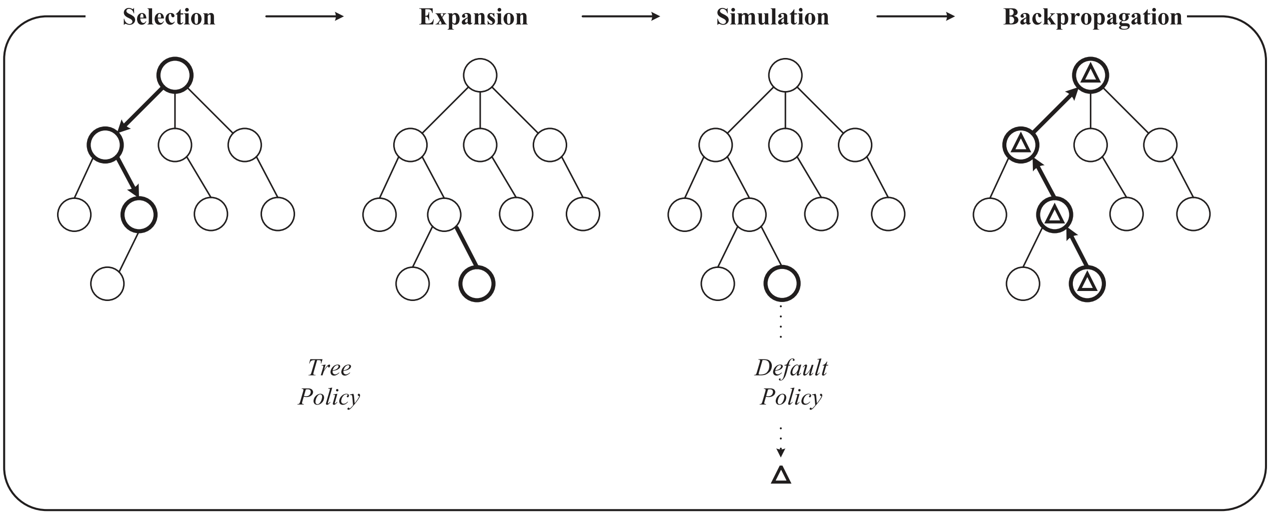


FIGURE 3
The traditional MCTS in one iteration. This process starts from a root node. Tree Policy is used to select feasible nodes. Default Policy or Rollout Policy is used to rapidly find the result in simulation. Finally, the result Δ is backpropagated to all nodes visited during this iteration.

a vector. It enables the quick search process of SL-MCTS to be more efficient than MCTS. The state value v is a scalar representing the path quality in each direction predicted at this position. It guides SL-MCTS toward the best-preferred strategy.

The pipeline of SL-MCTS is shown in Figure 4. It includes four steps: Selection, Expansion, Evaluation and Backpropagation. Suppose that at time step t, the agent is at node m_t. Regard m_t as the root node. One iteration of SL-MCTS at time step t is as follows:

1. Selection. If m_t is not a leaf, the agent uses Tree Policy to descend through the search tree until the most urgent expandable node is found. The Tree Policy of SL-MCTS is represented by Eqs (1) and (2). Equation (1) balances between exploitation ($\bar{Q}(x)$) and exploration ($U(s_t, x)$) of search.

$$m_{next} = \arg\max_{x \in AE(m_t)} (\bar{Q}(x) + U(s_t, x)) \tag{1}$$

$$U(s_t, x) = \frac{C_{puct} P(s_t, x) \sqrt{\ln N(m_t)}}{1 + N(x)} \tag{2}$$

where m_t is the location of agent in the search tree; s_t is the environment information at m_t; x is the child of m_t, $x \in AE(m_t)$; $P(s_t, x)$ is the selection probability of each child node x and is one of the predictions of PV-Network; $AE(m_t)$ is a set of legal action for m_t; $N(m_t)$ is the visit count of m_t; $N(x)$ is the visit count of x; $C_{puct} > 0$ is a hyperparameter, which means the amount of exploration performed; m_{next} is the branch selected for further exploration.

2. Expansion. If m_t is a leaf node, the available neighbor node(s) $AE(m_t)$ are added to expand the search tree.
3. Evaluation. PV-Network predicts the state value v and the selection probability $\boldsymbol{p}$ in the iteration.
4. Backpropagation. The visited count and action value Q are backpropagated through the search tree to update nodes'

```
Output: PV-Network model $f_\theta$
Initialization: Map information, PV-network $f_\theta$ and other parameters.
repeat
  while termination-condition-not-met do
    $state' \leftarrow state$
    for Iter ←0, MaxIteration do
      $v_l \leftarrow TreePolicy$
      $(p, v) \leftarrow$ PV-Network($state_{v_l}$)
      Backup the visited count $N$ and the action value $Q$
    end for
    $action$, $searchprob$ ← Select-action-by-visited-number ($state'$)
    $state$ ← Interact-with-the-environment ($action$)
  end while
  score $z$ ← Evaluating-with- optimal-historical-model ($path$)
  Output Dataset $(s_t, searchprob, z)$
  $value$, $selectionprob$ ← Prediction-by-PV-network ($s_t$)
  $loss$ ← Loss-function ($searchprob$, $selectionprob$, $z$, $value$)
  Updating PV-network parameters $f'_\theta$
  if $f'_\theta$ better than $f_\theta$ in tournament then
    Recording $f'_\theta$ as the optimal historical model.
  end if
until end
```

Algorithm 1. SL-MCTS path planning algorithm.

statistics. The Q value corresponds to the aggregate reward of all rollouts that pass through this state. The statistics are update by Eqs 3) and (4):

$$N(m_n)' = N(m_n) + 1 \tag{3}$$

and

$$\bar{Q}' = \frac{N(m_n) \times \bar{Q} + v}{N(m_n)'} \tag{4}$$

where m_n is one node in the search tree. $\bar{Q}$ is the action value of m_n before it is updated; $\bar{Q}'$ is the value after it updates; $N(m_n)$ is the visited number of m_n; The state value v is one of the outputs of PV-network.

When the iteration limit has been reached, the next move m_{t+1} is selected from node m_t based on the search probability π of SL-MCTS:

$$\pi(a|s_t) = \arg\max_a \frac{N(a)}{N(m_t)}, a \in AE(m_t) \tag{5}$$

where a is the child node of m_t; $N(a)$ is the visited count of node a.

SL-MCTS differs from the Simulation phase of the traditional MCTS algorithm. PV-Network replaces the rollout process in the traditional MCTS algorithm and can predict the selection probability of the feasible nodes and the state value. SL-MCTS has a more efficient search process and a more accurate search direction.

2.3.1. PV-network

The architecture of PV-Network is shown in Figure 5. PV-network consists of a backbone and then is divided into a policy branch and a value branch to output the selection probability $\boldsymbol{p}$ and the state value v. The backbone consists of three convolutional layers, and the kernel size is 3 × 3 with stride one and activated by the ReLU function. This network utilizes the convolutional layers to extract local information on the map, followed by fully connected layers to extract global information. The number of channels of these three convolutional layers in the backbone is 32, 64, and 128, respectively. The output of the backbone is used as input to the policy branch and value branch. The policy branch outputs a vector $\boldsymbol{p}$. The value branch outputs a scalar, v.

Figure 6 represents transforming from map information to the input features of PV-Network. The size of input S_t is $n \times n \times 4$ where $n \times n$ is the map size. The input comprises four binary feature matrices. The first matrix represents the start position of the task (Figure 6, Layer 0); the second represents the end position (Layer 1); the third represents the position of all obstacles on the map (Layer 2); the fourth represents the position of the nodes on the historical route (Layer 3). The four metrics are represented by "1" for existence and "0" for non-existence. For example, in Layer 3 in Figure 6, the node on the path is noted as "1" and the other as "0." $\boldsymbol{p}$ is a vector including the probability of the feasible nodes at S_t. The state value is a scalar in the range of $(0, 1)$.

2.3.2. The framework of self-learning

Self-learning is the process of SL-MCTS generating data for training and gradually improving its decision-making ability by learning those data. Firstly, the initial model is recorded as the optimal historical model. Then, the quality of SL-MCTS's solutions is evaluated using the optimal historical model. A higher score is given to the solution of the current model if it is better than the existing model. As a result, the current model is recorded as the optimal historical model, and the optimal historical mode is generally updated during the training. The data for model training is generated based on the solutions and scores. Repeating the above process, SL-MCTS improves its ability to find the optimal path and generates better training data.

The detail of the self-learning framework is shown in Figure 7. The beginning and destination of the task represent m_0 and m_E, and the parameters of the PV-Network f_θ are denoted by θ. The initialization state of each task is noted as s_0. The Evaluation process of SL-MCTS makes sampling based on the predicted selection probabilities $\boldsymbol{p}$ and the state value v by the network f_θ. Then, SL-MCTS selects a node m_1 to move and transfer from s_0 to s_1. The search finishes until the endpoint x_d is reached. As shown in Figure 7, SL-MCTS generates a path *path*. The quality of its path is evaluated by the result of the optimal historical model to get a score z. The optimal historical model is the best model based on the evaluation method of the Elo rating system (details in Section 3.2) during the training process. SL-MCTS with the optimal historical model produces a result of $path_b$. The path score is calculated depending on Eqs (7) and 8). *path* is split into data of the format $(s_t, \boldsymbol{p}_t, z)$ based on the number of nodes. These

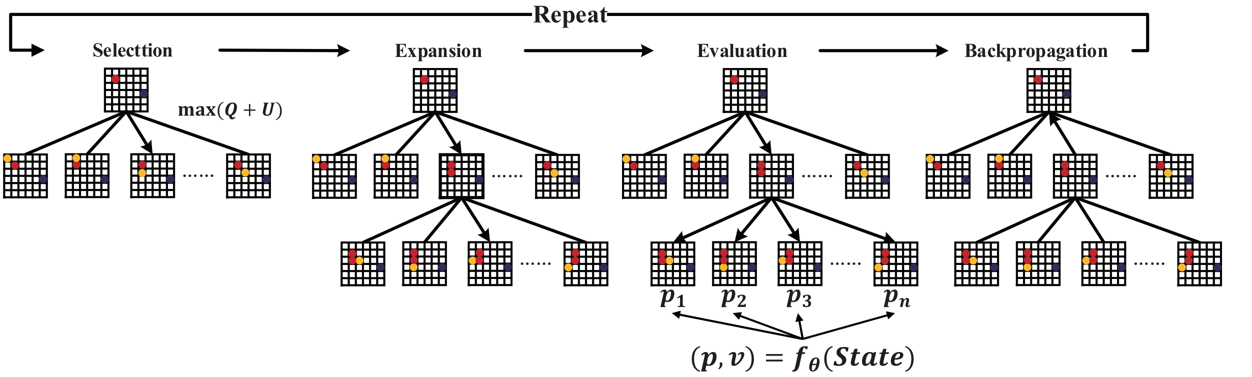


FIGURE 4
The pipeline of SL-MCTS. f_θ is PV-Network. The state is the input of the neural network. The output is the selection probabilities $\boldsymbol{p}$ of each child node and the state value v. The deep blue node indicates the endpoint, the red node indicates the historical route during the search, and the yellow node indicates the feasible space under the current state.

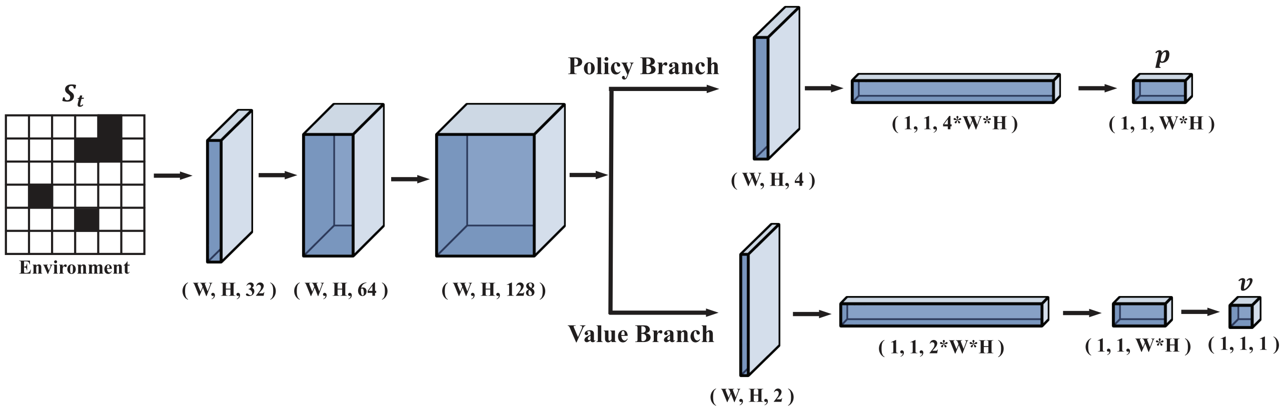
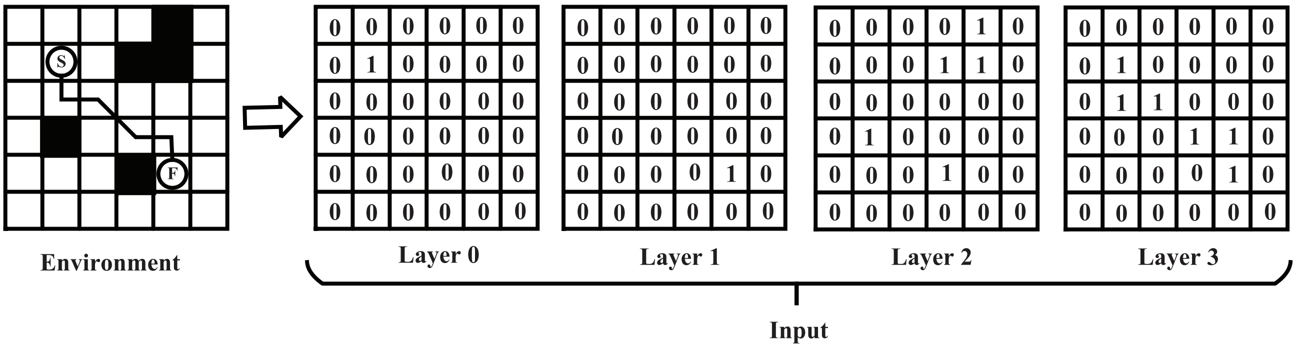


FIGURE 5
The architecture of PV-Network. W is the width of the map and H is the height of the map. $\boldsymbol{p}$ is the output of the policy branch and v is the output of the value branch.



Layer 0

0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Layer 1

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0

Layer 2

0	0	0	0	1	0
0	0	0	1	1	0
0	0	0	0	0	0
0	1	0	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0

Layer 3

0	0	0	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
0	0	0	1	1	0
0	0	0	0	1	0
0	0	0	0	0	0

FIGURE 6
Transformation process from map information to input features.

data are independent and are stored in the training data set. In the training process, SL-MCTS solves many random tasks and generates data. Lastly, PV-network is trained by randomly sampling the training data set in a small batch. This method of splitting data

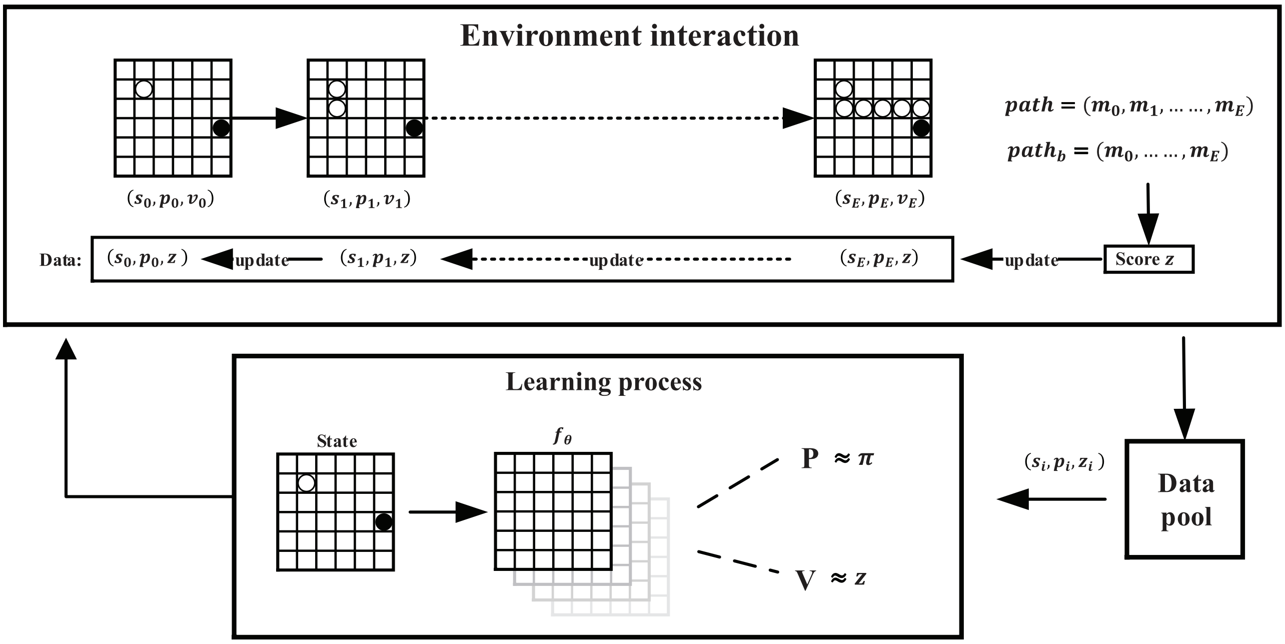


FIGURE 7
Training pipeline of SL-MCTS algorithm. path is the result of SL-MCTS algorithm in the training process. $path_b$ is the path planning result of SL-MCTS with the optimal historical model, which is used to assess the path score of *path*. The outcome of the assessment is recorded as path score *z*. The historical experiments are saved in the training data set. The beginning and destination positions of *path* and $path_b$ are the same, but the path lengths may differ. m_0 and m_E are the beginning and destination of the example task in this figure. SL-MCTS generates training data by solving many tasks with different beginning and destination positions.

can significantly break the association between paths and improve the algorithm's stability.

The loss function of PV-Network is:

$$loss = (z - v)^2 - \boldsymbol{\pi}^T \log \boldsymbol{p} + c \|\theta\|^2 \tag{6}$$

where c is a hyperparameter controlling the level of L2 weight regularization, which is to prevent overfitting and controls the contribution of the regularization term to the loss function. The network parameters θ are adjusted based on the loss function Eq. (6) to minimize the error between the predicted state value v and path score z and to maximize the similarity between the selection probability $\boldsymbol{p}$ and the search probability $\boldsymbol{\pi}$.

To expand the range of exploration of SL-MCTS in the training process and avoid falling into the local optimal trap, Dirichlet noise is added to the selection probability $p(s, a) \leftarrow (1-\varepsilon_1)p(s, a)+\varepsilon_1\eta_a$, where s is the state, a is legal action, and $p(s, a)$ is the predicted selection probability of each a. ε_1 is set to 0.5, and it is used to encourage the exploration of different actions. Dirichlet noise is also added into the search probability $\pi \leftarrow (1-\varepsilon_2)\pi+\varepsilon_2\eta_a$, where $\eta_a \sim Dir(0.3)$ and ε_2 is 0.25, to encourage SL-MCTS to explore every feasible node during the training process. The higher ε_2 is, the more different states are explored and thus enhance the data diversity of the PV-Network.

In the path planning task, the path evaluation is not only related to whether the endpoint is reached but also considers the length of the path. Using only Euclidean distance or Manhattan distance is not reasonable to evaluate path quality. This method can not reflect the existence of obstacles on the line between two points and provides the agent with ambiguous feedback that does not reflect changes in the quality of its solution. Therefore, SL-MCTS generates a path score representing the current problem-solving ability by comparing their results with the optimal historical model. The path score is given by Eqs (7) and (8):

$$l = len(path_b) - len(path) \tag{7}$$

$$z = \frac{2}{1 + e^{-\gamma l}} - 1 \tag{8}$$

where $\gamma \in (0, 1]$. If the result of Eq. (7) is <0, it denotes that the solution of the optimal historical model is better than the solution of the current model. *path* receives a score under zero, which means that similar decisions are discouraged. In contrast, if the result of Eq. (7) exceeds 0, indicating that the path length of the optimal historical model is longer than that of the current model, *path* receives a score above zero, which means that those similar decisions are encouraged. Furthermore, if SL-MCTS with the current model fails to reach the destination, this *path* receives a score, -1. The evaluator of SL-MCTS is dynamically adjusted according to the update of the optimal historical model during the training process.

2.4. Computational complexity

As there are many different tasks in path planning, it is difficult to assess the computational complexity accurately. The computational complexity of SL-MCTS is analyzed by referring to the calculation method in Yonetani et al. (2021) and Qi et al. (2021). The difference in computational complexity between SL-MCTS and MCTS is mainly in the simulation phase at each time step. Therefore, the analysis focuses on the differences in their computational complexity during the simulation phase. Suppose the length of the path is l, a is the feasible space for each node, and k is the number of simulations per search process. For the traditional MCTS algorithm, the maximum search depth in the rollout process is d, and its computational complexity is denoted as $\mathcal{O}(lk(ad))$. The computational complexity of PV-Network is defined as $\mathcal{O}(|V|)$ in the training process, according to Yonetani et al. (2021). After training, the computational complexity of the SL-MCTS inference phase is $\mathcal{O}(lka)$ and $\mathcal{O}(lk)$ for worst and best cases.

3. Experiments and analysis

This section provides detailed descriptions on the experimental settings, parameter adjustments, and evaluation methods. We conducted the training process of SL-MCTS on maps with different scales and analyzed the variability of its problem-solving capability. Additionally, we compared the performance of SL-MCTS with other advanced single-player MCTS algorithms and collective intelligence algorithms. Furthermore, we verified the generalization of SL-MCTS on random layout maps with specific obstacle densities and the dynamic environmental map. Finally, we conducted ablation experiments to explore the impact of different simulation times on SL-MCTS. The open-source code, experimental data, and detailed visualizations of the experimental data and results can be found in Liu (2023).

3.1. Experimental settings

These experiments were implemented in Python 3.7 using PyTorch. They were executed on a high-performance computing server, using two GeForce RTX 2080 SUPER GPUs for algorithm training in parallel and CPUs that are 3.20 GHz with 16GB memory. The number of simulations of SL-MCTS is set to 30 and C_{puct} is $1/\sqrt{2}$. The Adam optimizer optimizes the neural network. The learning rate is 10^{-3}, and its initial multiplier (lr_m) is 1.0. To avoid updating the policy parameters too much at each training iteration, the KL divergence (Nielsen, 2020) is used to adjust lr_m to improve the training stability. Referring to the Proximal Policy Optimization algorithm (Schulman et al., 2017), the probability distributions generated before and after policy updating (p_{old} and p_{new}) are used to calculate their KL divergence based on the result of Eq. (9). lr_m is adjusted by Eq. (10).

$$KL(p_{old} \parallel p_{new}) = \sum p_{old} \cdot log \frac{p_{old}}{p_{new}} \tag{9}$$

$$lr_m = \begin{cases} 1.5 \cdot lr_m, & if\ KL < \frac{kl_{targ}}{2}\ and\ lr_m < 10 \\ \frac{lr_m}{1.5}, & if\ KL > 2 \cdot kl_{targ}\ and\ lr_m > 0.1 \end{cases} \tag{10}$$

where the parameter kl_{targ} is 0.02.

In order to investigate the performance of SL-MCTS on environmental maps of varying scales, we conducted experiments on 6 × 6 and 16 × 16 maps, respectively. The size of the training data set is 10,000. If the data set is completely full, older data is automatically removed as newer data are added Positive samples are defined as those paths that reach the destination and achieve equal to or shorter lengths than the optimal historical model's results. To provide a high-quality training data set for the initial training process of SL-MCTS and rapidly promote the ability of SL-MCTS, the positive sample and negative sample is stored by a $1:1$ ratio in the training data set at the initial stage of training.

In this paper, SL-MCTS algorithm compares with variants of MCTS like UCB1 (Auer et al., 2002), MCTS (or UCT) (Kocsis and Szepesvári, 2006) and the variations of SP-MCTS (such as those presented in Schadd et al., 2012; Crippa et al., 2022), to verify its performance. SP-MCTS-CRIPPA (Crippa et al., 2022) is one of the best single-player MCTS algorithms. Additionally, this paper compares SL-MCTS algorithm to prevailing collective intelligence algorithms, including ACO algorithm (Dorigo et al., 2006) and PPACO algorithm (Luo et al., 2020). PPACO is an improved ACO algorithm for path planning problems, which is one of the best ACO algorithms for solving path planning. It has domain-specific knowledge.

3.2. Evaluation method

Elo rating system (Coulom, 2008) is used to evaluate the variation of SL-MCTS's problem-solving ability in the training process. The initial Elo ratings of algorithms are 1,000. MCTS-50 and MCTS-150 (Kocsis and Szepesvári, 2006) were chosen for comparison with SL-MCTS, where the number of them denotes the number of simulations. The solution of MCTS-150 is generally better than that of MCTS-50 because the MCTS algorithm can improve its problem-solving capabilities by increasing the number of simulations and the depth of exploration. In this paper, we define the case where SL-MCTS finds the destination, and the path is shorter than the competitor as a win; the case where it finds the destination but the path length is the same as the competitor as the tie; otherwise, it is considered as the failure. The two algorithms update their rating by a "shorter path finding" tournament, which consists of 100 different random tasks. The details of updating the rating are as follows. The expected score of player a is presented as

$$E_a = \frac{1}{1 + 10^{\frac{R_b - R_a}{400}}} \tag{11}$$

and the expected score of player b is

$$E_b = \frac{1}{1 + 10^{\frac{R_a - R_b}{400}}}, \tag{12}$$

where R_a is the rating of player a. After the tournament, if the actual rating of player a (S_a) differs from its expectation of E_a, the level R_a is adjusted as follows:

$$R_a^{'} = R_a + K(S_a - E_a), \tag{13}$$

where K is the hyperparameter, which means the range of changes in Elo rating. In this paper, the algorithm's high rating means that it wins more times than its opponent in the tournament, i.e. most of its path lengths are shorter than its opponent's.

To assess the performance of SL-MCTS on the path planning problem, we compared the average path length, time consumption, the standard deviation of path lengths (SD-L) and time consumption (SD-T), visited range and the percentage of successfully solved tasks (Success rate). A smaller average path length reflects a better solution quality of the algorithm. Average time consumption reflects the algorithm's efficiency in solving problems. SD-L and SD-T reflect the variation of the algorithm in the quality and efficiency of solutions. The visited range represents the ratio between the number of visited nodes and the total number of feasible nodes in the map. The success rate is defined as the proportion of successfully completed tasks to the total number of tasks and serves as one of the criteria of the algorithm's problem-solving performance. We also employed the Mann-Whitney U test as a significance test to determine the mean difference between the experimental results for algorithms. The significance level is set to 0.05.

3.3. Results and discussion

3.3.1. Performance of self-learning

SL-MCTS's self-learning performances in two scale environmental maps are respectly present. One hundred tasks with different origins and destinations are randomly selected as a tournament from each environment. We used the Elo rating to illustrate the variation of SL-MCTS's problem-solving ability. The initial rating of the Elo rating system (Detailed in Section 3.2) is set to 1,000.

Figure 8A shows the Elo rating curves of SL-MCTS, MCTS-50 and MCTS-150 in an obstacle-free 6 × 6 environmental map. Figure 8B shows the performance of SL-MCTS in the 16 × 16 map, which includes 211 feasible nodes and 45 obstacle nodes (as shown in Figure 10). As traditional MCTS (Kocsis and Szepesvári, 2006) has no ability to learn the history experiment, its Elo rating is not changed. As shown in Figure 8A, the Elo rating score of MCTS-150 is 1,234, while that of MCTS-50 is 766. In contrast, SL-MCTS algorithm has a considerably lower rating of 680 before any training has taken place, in contrast to the other two traditional MCTS algorithms. At the 1*th* evaluation in the training process of the SL-MCT, the rating of SL-MCTS is 904, which is higher than MCTS-50. At the 7*th* evaluation, its Elo rating is 1,240, which has already exceeded MCTS-50 and MCTS-150. These results indicate that the problem-solving capability of SL-MCTS in the 6 × 6 map is better than MCTS algorithms at 7*th* evaluation. Eventually, the Elo rating of SL-MCTS is 1,368. This value is approximately twice the original Elo rating of the SL-MCTS. As shown in, Figure 8B, the Elo rating of MCTS-50 is 712, and the Elo rating of MCTS-150 is 1,288. The Elo rating of the SL-MCTS algorithm is 576 before the training process, much lower than MCTS. The Elo rating of the SL-MCTS at the 1*th* evaluation exceeds the rating of MCTS-50, which is 760. At the 3*th* evaluation, SL-MCTS's Elo rating is 1,280, which is much similar to that of MCTS-150. The rating of SL-MCTS exceeds that of MCTS-150 at the 6*th* evaluation. The Elo rating of SL-MCTS finally reaches 1,632, which is almost triple the initial rating of SL-MCTS. These results show that the performance of SL-MCTS in the 16 × 16 map is better than MCTS algorithms at 6*th* evaluation. In conclusion, the experimental results in Figure 8 indicate that SL-MCTS performs much worse than MCTS-50 in the beginning (the maximum difference in their Elo rating is 136), which indicates SL-MCTS's initialized PV-Network cannot compete with the rollout process of conventional MCTS. Through self-learning, the Elo rating of SL-MCTS exceeds that of MCTS-50 at the first evaluation in both size environmental maps and exceeds that of MCTS-150 at about the seventh evaluation. Finally, after several training iterations, the Elo rating of SL-MCTS increased approximately three-fold from its initial Elo rating. This also implies that the performance of SL-MCTS significantly enhanced via the self-learning process. The experimental results suggest that SL-MCTS, guided by the PV-Network, can navigate toward a more efficient direction in comparison to the traditional MCTS's rollout process, ultimately leading to better solutions.

To further verify the variation of SL-MCTS's path-finding capability in the self-learning process, we randomly selected 50 tasks in the 6 × 6 map as a test set and compared the average total path length of SL-MCTS at different training stages with that of MCTS-50 (as shown in Figure 9). For each algorithm, the experiments were conducted five times on the test set, using the same parameters. The average of these experiments was used to determine the average total path length ($path_{at}$) of algorithm, which is calculated by:

$$path_{at} = \frac{1}{5}\sum_{i=1}^{5}\sum_{j=1}^{50} len^{ij}, \qquad (14)$$

where i represents the times of repeated experiments, while j denotes the number of tasks within the test set. $path_{at}$ for MCTS-50 is 152. Figure 9 illustrates the $path_{at}$ values generated by SL-MCTS at various learning stages. During the second training iteration, SL-MCTS generated an $path_{at}$ value of 134, which is comparatively shorter than that of MCTS-50. The $path_{at}$ value of SL-MCTS shows a decreasing trend as the number of training iterations increase. In particular, the final $path_{at}$ of SL-MCTS compared to that in the second training iteration decreased by 26%. The evidence of Figures 8, 9 implies that SL-MCTS has significantly improved its path-finding capacity through the process of self-learning.

In order to further investigate the guiding role of the PV-network during the reasoning process of SL-MCTS, the predicted probability results of feasible action selection in each step of SL-MCTS were analyzed in this section. As shown in Figure 10, the number on the map means the predicted probability and guides the search direction. "S" represents the start and "E" represents the destination of the task. "C" represents the position of the agent in that state. Figures 10A–C present the three states of the two tasks in 6 × 6 map. Figures 10D–F present the three states of one task in 16 × 16 map. Figure 10A shows that the probability of nodes close to the side of node E is significantly higher than nodes far from node E. The selection probability of node (3, 0) is 0.86, the highest value at that state. Figures 10B, C are the two states of another task which starts at (0,1) and ends at (2,4). The agent starts from node S in Figure 10B, and the agent is at node C in Figure 10C.

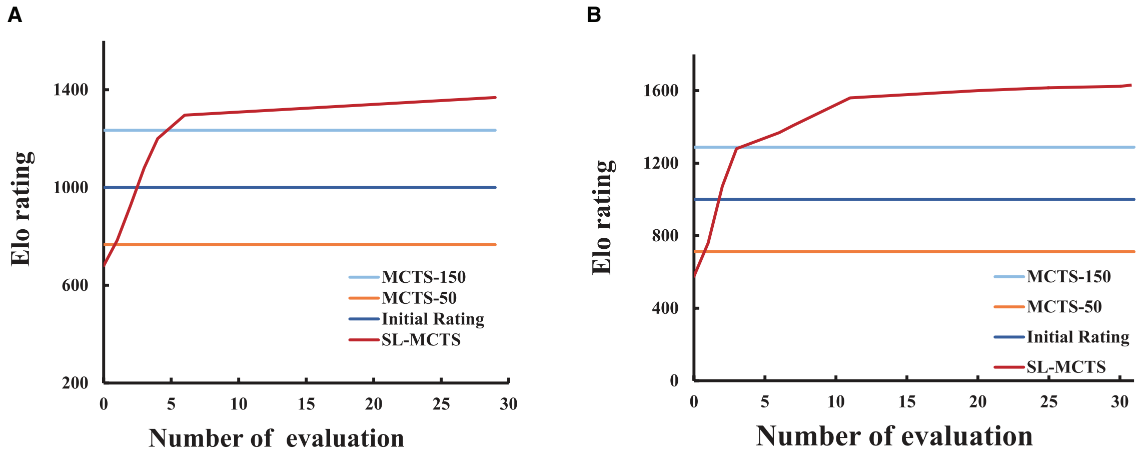


FIGURE 8
The Elo rating curves of SL-MCTS algorithm. **(A)** Represents the Elo rating curve for SL-MCTS in the 6 × 6 map. **(B)** Represents the Elo rating curve for SL-MCTS in the 16 × 16 map.

In Figure 10B, the maximum probability value is 0.79 at the node (1, 2). The agent executed action (1, 2) and transferred to the next state, as shown in Figure 10C. The node (2, 3) has the highest probability of 0.78 in this stage. The prediction results of SL-MCTS (shown in Figures 10A–C) all present that the nearest node to the destination has the highest selection probability. Figures 10D–F show the three states of one task in the 16 × 16 map, which starts at node (2, 6) and ends at node (14, 14). In Figure 10D, the agent starts from node S. Node (3, 7) has the highest selection probability of 0.5. The agent in Figure 10E is at point C, and the selection probability of (10, 14), which is closest to (14, 14), is the highest, and others are low; Figure 10F is the next state of Figure 10E, where the selection probability of node (11, 15) is the highest. The results in Figure 10 show that the well-trained PV-Network can provide a reasonable selection probability for SL-MCTS based on the global information of the map environment and the current location.

3.3.2. Comparative experiments

SL-MCTS is compared with UCB1 (Auer et al., 2002), MCTS (Kocsis and Szepesvári, 2006), SP-MCTS (Schadd et al., 2012), and SP-MCTS-CRIPPA (Crippa et al., 2022) to show its performance in path planning. The comparison algorithms include UCB1-50, UCB1-150, MCTS-50, MCTS-150, SP-MCTS-50, SP-MCTS-150, SP-MCTS-CRIPPA-50, SP-MCTS-CRIPPA-150 where the numbers indicated the number of simulations. SL-MCTS is also compared with the prevailing collective intelligence algorithm, ACO (Dorigo et al., 2006) and PPACO (Luo et al., 2020). The comparison algorithms included ACO-15-15, and ACO-30-30, where the numbers indicate the number of populations and iterations of ACO. The parameters of the ACO algorithms are set as follows: $\alpha = 1$, $\rho = 0.3$, $\beta = 1$. We chose three tasks with different origins and destinations: (1, 0) to (8, 0), (2, 14) to (7, 3), and (14, 2) to (6, 15). The span of the tasks' beginning and destination is increasing, which means that the task's difficulty is increasing. This is because, for the algorithm, a larger task span means that it needs to explore a wider area and potentially deal with more obstacles, making it more challenging to search for the destination. The algorithms' shortest path length (Best) in the fifty times of repeated testing, results of the average path length, the average time consumption, the visited range, the standard deviation of path length (SD-L) and time consumption (SD-T), the success rate of finding the destination and p-value after executing the task 50 times are shown in Table 1.

Table 1 shows that, for the traditional MCTS algorithms, in Task 1, UCB1-50 have the shortest optimal path of 10, with the shortest average path length of 14.1. In Task 2, SP-MCTS-CRIPPA-150 obtained an optimal path length of 12 and a shortest average path length of 15.5, but its success rate in solving problems is 0.94. In Task 3, UCB1-150 has an optimal path of 26 than other traditional MCTS algorithms and an average path length of 48.08. It's worth mentioning that the time consumption of the traditional MCTS algorithm increases significantly as the iteration times increase. For collective intelligence algorithms, ACO-30-30 has the smallest optimal solution and average path length for Task1, at 10 and 10.68 respectively. For Task 2 and Task 3, PPACO-30-30 has the shortest average path length out of all ACO algorithms, which is 15.22 and 17.68 respectively. Compared to traditional MCTS algorithms and collective intelligence algorithms, SL-MCTS-30 only explored 14.69% of the environment in Task 1 and it takes an average of 2.99 s. The quality of SL-MCTS-30's path is only inferior to ACO-30-30 and ACO-15-15. Its time consumption is the least. Its optimal path length is 10, with an average path length of 11.08. Furthermore, the standard deviation of SL-MCTS-30 in terms of path length and time consumption is the lowest among other compared algorithms, at 1.59 and 0.04 respectively. This suggests that the performance of SL-MCTS-30 is more stable. Mann–Whitney U-tests were performed to obtain the results between the algorithm with the best Average length (shown in bold italics) and other algorithms. In Task 1, ACO-30-30 is determined to be the best method. The results of the significance test show that there is no significant difference between AS-30-30 and SL-MCTS-30. In Task 2, the optimal

TABLE 1 Performance of UCB1, MCTS, SP-MCTS, SP-MCTS-CRIPPA, ACO, PPACO, and SL-MCTS algorithms on different tasks.

Algorithm	Best	Average length	Average time(s)	Visited range(%)	SD-L	SD-T	Success ratio(%)	p-value
Task 1								
UCB1-50	**10**	16.42	6.93	100	3.52	1.60	1	1.44×10^{-17}
UCB1-150	**10**	14.1	19.00	100	2.22	3.50	1	1.03×10^{-5}
MCTS-50	12	16.92	4.16	100	3.43	0.90	1	2.26×10^{-3}
MCTS-150	**10**	14.41	18.33	100	2.59	3.59	0.99	2.12×10^{-3}
SP-MCTS-50	12	19.08	8.35	100	8.34	5.09	1	1.12×10^{-11}
SP-MCTS-150	12	18.84	24.41	100	4.00	6.63	1	5.93×10^{-24}
SP-MCTS-CRIPPA-50	11	15.3	6.54	100	3.22	1.45	0.92	8.33×10^{-15}
SP-MCTS-CRIPPA-150	12	15.5	21.48	100	2.17	3.49	0.94	2.73×10^{-24}
ACO-15-15	11	99.52	11.00	6.24	**0.52**	0.68	1	3.99×10^{-5}
ACO-30-30	**10**	***10.68***	10.05	98.10	0.69	0.94	1	–
PPACO-30-30	11	11.58	6.24	99.22	1.04	0.68	1	0.007
SL-MCTS-30	**10**	**11.08**	**2.99**	**14.69**	0.59	**0.04**	1	$\mathbf{1.85 \times 10^{-1}}$
Task 2								
UCB1-50	17	28.2	13.04	100	5.96	2.92	1	$1.60 \times 10_{-26}$
UCB1-150	17	24.78	41.16	100	5.26	10.96	1	$3.13 \times 10_{-21}$
MCTS-50	16	39.08	17.59	100	11.04	6.67	1	2.92×10^{-27}
MCTS-150	16	38.94	64.97	100	9.13	20.66	1	8.04×10^{-33}
SP-MCTS-50	**12**	19.08	8.35	100	8.34	5.09	1	6.15×10^{-4}
SP-MCTS-150	**12**	18.84	24.41	100	4.00	6.63	1	7.03×10^{-10}
SP-MCTS-CRIPPA-50	30	42.36	20.75	100	9.18	4.91	0.76	1.61×10^{-20}
SP-MCTS-CRIPPA-150	**12**	15.5	21.48	100	2.17	3.49	0.94	2.82×10^{-22}
ACO-15-15	15	17.66	10.46	99.52	1.19	**0.48**	1	3.39×10^{-12}
ACO-30-30	14	16.46	61.57	99.52	**0.98**	0.69	1	9.16×10^{-8}
PPACO-30-30	13	***15.22***	60.84	99.52	1.86	2.24	1	–
SL-MCTS-30	13	16.20	**5.34**	**19.90**	3.12	1.17	1	$\mathbf{2.08 \times 10^{-2}}$
Task 3								
UCB1-50	42	59.27	29.46	100	10.62	6.02	0.98	2.36×10^{-47}
UCB1-150	26	48.08	94.42	100	9.74	23.47	1	2.65×10^{-39}
MCTS-50	33	56.85	28.89	100	13.03	6.76	0.96	8.61×10^{-38}
MCTS-150	26	47.24	65.84	100	9.18	12.71	1	2.26×10^{-40}
SP-MCTS-50	42	73.16	42.66	100	19.97	17.76	1	3.66×10^{-44}
SP-MCTS-150	42	73.16	42.66	100	19.97	17.76	1	1.30×10^{-35}
SP-MCTS-CRIPPA-50	–	–	–	–	–	–	–	–
SP-MCTS-CRIPPA-150	84	84	172.21	100	–	–	0.04	–
ACO-15-15	17	20.70	**16.62**	90.05	1.75	**1.08**	1	1.65×10^{-14}
ACO-30-30	**16**	18.9	82.97	98.52	**1.45**	3.09	1	5.67×10^{-5}
PPACO-30-30	**16**	***17.68***	69.46	99.60	1.55	2.19	1	–
SL-MCTS-30	18	42	24.48	**28.90**	21.89	14.13	0.92	2.82×10^{-14}

The best value for each evaluation metric is marked in bold.

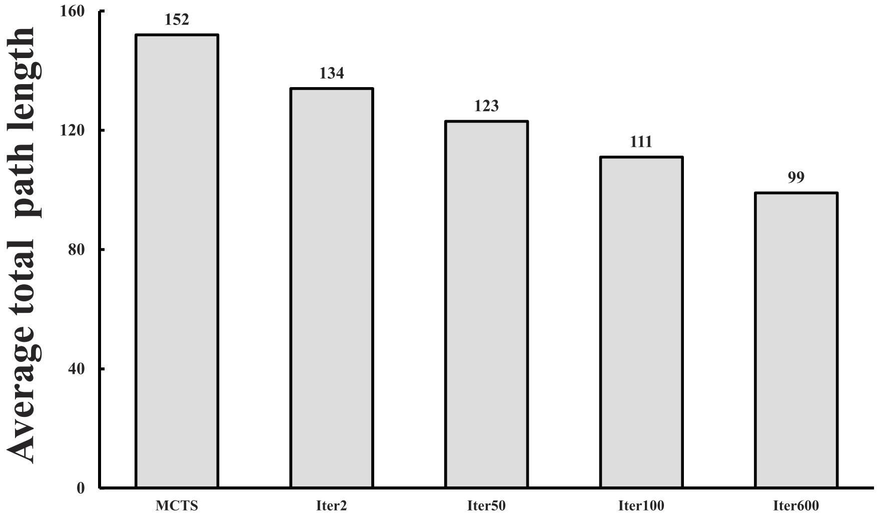


FIGURE 9
The average total path length of SL-MCTS in different training stages (as shown in columns 2, 3, 4, 5) compared with that of MCTS-50 (As shown in column 1).

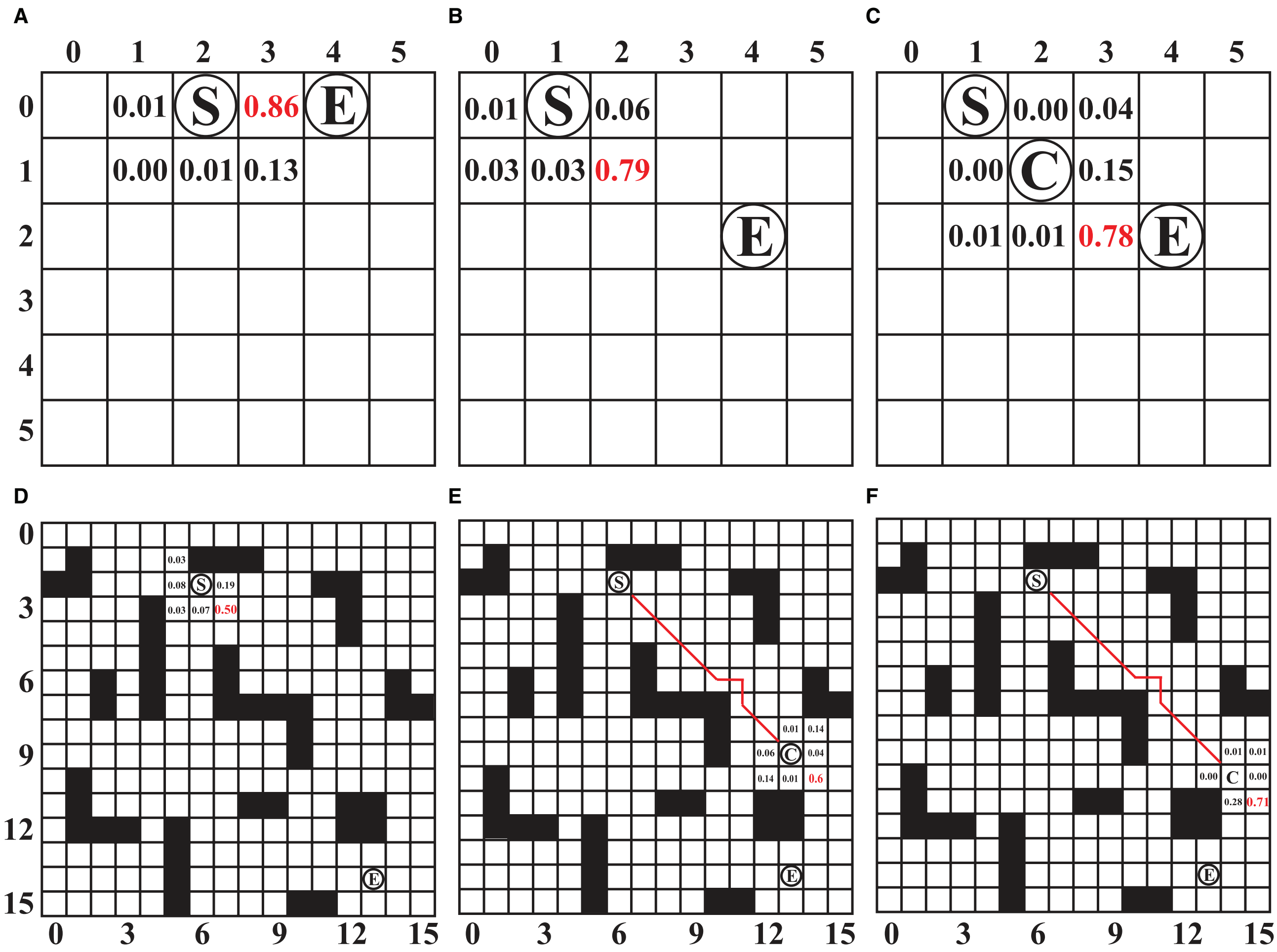


FIGURE 10
(A–F) The predicted selection probability of SL-MCTS for different states. "S" represents the start and "E" represents the destination of the task. "C" represents the position of the agent in the state. Where the value of the number represents the predicted selection probability value of SL-MCTS. The number in red indicates the position with the highest value in the predicted result.

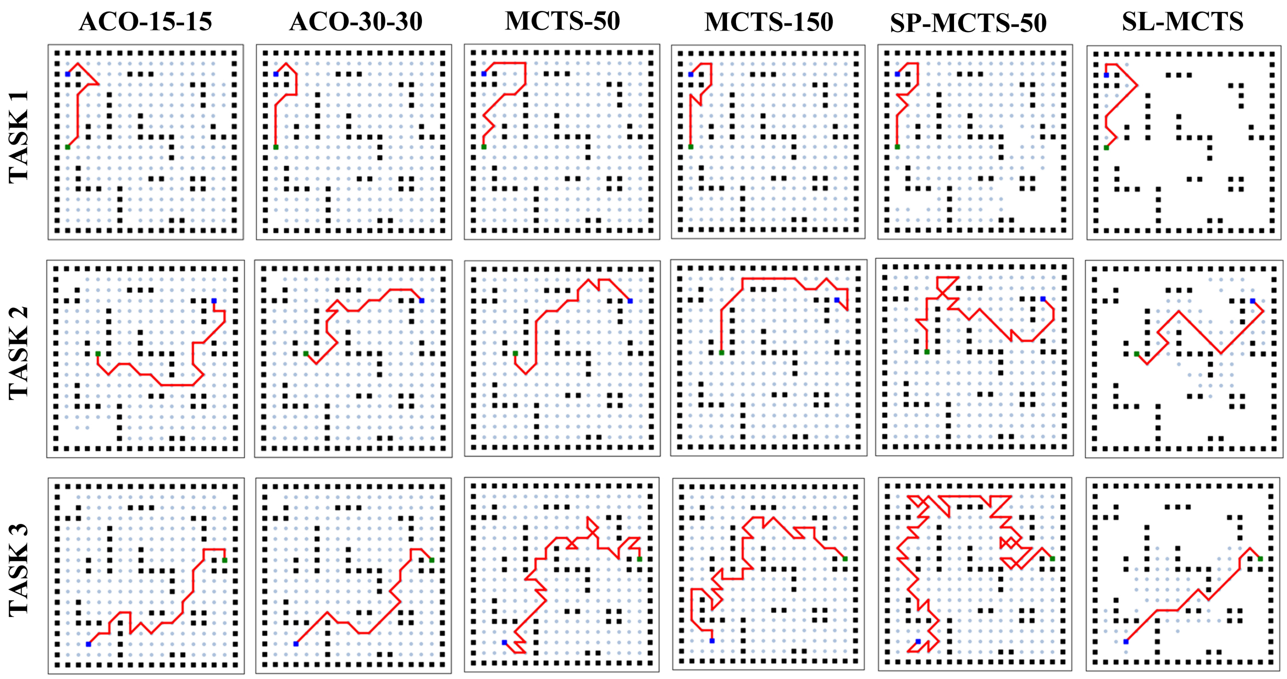


FIGURE 11
Visualization of path planning results for ACO, MCTS, and SL-MCTS. Black nodes indicate obstacles. Blue nodes indicate origins, green nodes indicate destinations and red lines indicate the found paths. The range visited by the algorithm is marked with light blue nodes.

solution of SL-MCTS-30 is 13, which is the same as PPACO-30-30, second only to SP-MCTS and SP-MCTS-CRIPPA-150. The average path length of SL-MCTS is 16.20, only 0.7 longer than that of SP-MCTS-CRIPPA-150 and PPACO-30-30, but the exploration space of SL-MCTS is only 19.9%, one fifth of other algorithms. Additionally, the average consumption time for SL-MCTS was the shortest amongst all algorithms, taking only 5.34*s*. PPACO-30-30 is determined to be the best method. The results of the significance test show that there is no significant difference between PPACO-30-30 and SL-MCTS-30. In task 3, the optimal solution of SL-MCTS-30 is 18, which is only second to the ant colony algorithms. Moreover, SL-MCTS explores only 28.90% of the environment space and solves the problem in just 25.48 s, making it a highly efficient algorithm. In conclusion, SL-MCTS with a simulation count of 30 performed significantly better than traditional MCTS and SP-MCTS algorithms with simulation counts of 50 or 150. Its performance is comparable to that of ACO, which is proficient at solving path planning tasks. The experimental results show that under the guidance of the PV-Network, SL-MCTS converges faster than other MCTS algorithms. However, SL-MCTS is considerably more efficient than ACO in terms of time consumption and search space for most tasks, with time consumption of less than half and search space only one fifth that of ACO. It is meaningful to mention that some MCTS algorithms are unable to solve the complex path planning problem (such as SP-MCTS-CRIPPA), mainly because most traditional MCTS algorithms are designed for game scenarios and not proficient at solving path planning tasks. However, with the proposed method in this paper, SL-MCTS has made significant improvements over MCTS algorithms

Figure 11 visualizes the planning results of SL-MCTS, MCTS-50, MCTS-150, SP-MCTS-50, ACO-15-15, and ACO-30-30. Black nodes indicate obstacles, blue nodes indicate origin, green nodes indicate destinations and red lines indicate found paths. The range visited by the algorithm is marked with light blue nodes. The visualization of these algorithms' path planning results presents that the paths of SL-MCTS have fewer inflection points than other traditional MCTS algorithms, and the number of visited nodes is much less than others. This also indicates that the search of SL-MCTS is efficient, and the path of SL-MCTS is reasonable and competitive with other baselines.

3.3.3. Generalization of SL-MCTS

This section aims to evaluate the potential of SL-MCTS in tackling tasks in previously unseen environmental maps. Two sets of experimental maps, each with three different obstacle densities, were constructed based on the two map sizes. Fifty tasks of random starting and ending points were selected on each map to form the test set of each map. The map size is 6 × 6, defined as MAP 1, and 16 × 16, defined as MAP 2. The number of SL-MCTS's simulations is 30. Maps are named Sparse Map 1, Moderate Map 1, Dense Map 1, Sparse Map 2, Moderate Map 2 and Dense Map 2 according to the density of obstacles in the maps (5%, 25%, and 55%). In Table 2, we presented the performance of SL-MCTS by analyzing the ratio of SL-MCTS to MCTS-50 in terms of path length ($Ratio_{length}$) and

TABLE 2 Comparative analysis of path lengths and time consumption for SL-MCTS and MCTS-50 in the test map sets.

	$Ratio_{length}$	$Ratio_{time}$	Success rate(%)
Sparse-Map 1	0.92	0.59	100
Moderate-Map 1	0.94	0.67	100
Dense-Map 1	0.91	0.55	100
Sparse-Map 2	0.84	0.54	100
Moderate-Map 2	0.82	0.58	100
Dense-Map 2	0.64	0.42	100

time consumption ($Ratio_{time}$), which were calculated by:

$$Ratio_{length} = \frac{1}{50}\sum_{i=1}^{50}\frac{path_{Ai}}{path_{Bi}} \tag{15}$$

$$Ratio_{time} = \frac{1}{50}\sum_{i=1}^{50}\frac{time_{Ai}}{time_{Bi}} \tag{16}$$

where *i* is the number of the testing tasks. *A* represents SL-MCTS and *B* represents MCTS-50. A lower ratio indicates that SL-MCTS performs better than MCTS-50 in terms of path quality or time consumption. The success rate is defined as the proportion of successfully completed tasks to the total number of testing tasks.

In Table 2, the success rates of SL-MCTS in maps are 100%. It means that SL-MCTS can successfully tackle the tasks in these unseen environmental maps. The $Ratio_{length}$ value is about 0.90 on the set of maps for MAP 1 and about 0.76 on that for MAP 2. The $Ratio_{time}$ value is about 0.5 both on maps 1 and 2. These experiments indicate that SL-MCTS performs significantly better than MCTS-50 in terms of path quality, particularly in environments with a map size of 16. Furthermore, SL-MCTS completes the same task using only half of the time computation required by MCTS-50. SL-MCTS performs better on MAP2 than on MAP1, which may be due to the larger search space and greater number and variety of obstacles on MAP2, making tasks more challenging and enabling SL-MCTS to demonstrate its superior capabilities. In general, these experiments demonstrated that SL-MCTS not only is able to find the tasks' solutions on the new maps but also completes them with half the time required by MCTS-50, particularly for tasks with shorter lengths.

We conducted additional experiments on random maps with different obstacle distributions. By comparing the proposed algorithm's performance in solving the same task in these diverse environmental maps, we further assessed SL-MCTS's ability to adapt to novel environmental maps. We chose two test tasks: one on a map with a size of 6, with a starting point at (0, 0) and an ending point at (5, 5); the other on a map of size 16 with a starting point at (3, 8) and an ending point at (14, 14). The considerable span of both tasks on their respective maps allowed us to examine different obstacle distributions. Tables 3, 4 display the results performed by SL-MCTS on different-sized maps, and these results are compared with those of MCTS-50. The "prior map" in these tables refers to the environmental map utilized for SL-MCTS learning, while the "random map" denotes an environment with a different obstacle distribution compared with "prior map," which SL-MCTS has unseen before. We have provided more information about the environmental map in the public code repository (Liu, 2023). The test tasks were repeated 50 times per map. This section analyzed the ability of SL-MCTS to handle tasks in new environments by comparing its best and average path lengths, the standard deviation of path lengths (SD-L), average time consumption (average time) and standard deviation (SD-T) of time consumption, and success rate with those of MCTS-50. We also employed the Mann-Whitney U test as a significance test to determine the mean difference between the experimental results for SL-MCTS and MCTS-50 (the best Average length, shown in bold italics).

According to the results in Table 3, SL-MCTS outperforms MCTS-50 in both the "prior map" and new "random map" environments. Specifically, SL-MCTS had a much shorter average path length than MCTS-50, along with a smaller standard deviation in path lengths. This indicates a higher solution quality and lower fluctuation compared to MCTS-50. In addition, SL-MCTS also consumed significantly less time on average than MCTS-50. Furthermore, the results of the significance test in both "random map1" and "random map2" show that there is a significant difference between SL-MCTS and MCTS-50, with SL-MCTS being the best method. Table 4 shows that SL-MCTS's average path length and SD-L in "random map" environments were similar to those of MCTS-50. SL-MCTS's success rate on "random maps" was 0.68 lower than that on the "prior map." This could be attributed to the excessive density of obstacle distribution between the start and end points, including an obstacle corridor that blocks access between the beginning and the destination. This significantly increases the difficulty of the testing task on the "random map" compared to that on the prior map. The results of the significance test in the "random map" show that there is no significant difference between SL-MCTS and MCTS-50. The results show that SL-MCTS can solve the tasks on the new maps, indicating that the problem-solving ability of SL-MCTS has generalization in unseen environmental maps.

3.3.4. Ablation experiments

This section presents the effect of PV-Network on the SL-MCTS algorithm with a different number of simulations. Thirty tasks are randomly selected from the 16 × 16 map as a test set. The variation of the total length and the total time consumption of SL-MCTS-30 and MCTS-30 was compared on the test set. As shown in Figure 12, five different simulations (10, 30, 50, 70, and 90) are chosen. The total path lengths of SL-MCTS and MCTS decrease as the number of simulations increases, which means that increasing the number of simulations can improve the quality of MCTS's solution. However, for different simulation numbers, SL-MCTS has significantly shorter path lengths than MCTS, being almost half of MCTS's lengths. Although the time consumption of both algorithms increases with the number of simulation, traditional MCTS algorithms become more time-consuming with higher simulation numbers. And the time consumption of SL-MCTS is consistently lower than MCTS, about two-fifths of MCTS's total time. Experiments show that PV-Network can provide accurate guidance for the search process of SL-MCTS, and SL-MCTS is more

TABLE 3 Results of SL-MCTS and MCTS on different 6 × 6 maps.

	Best	Average length	SD-L	Average time	SD-T	Success rate(%)	p-value
Prior map							
SL-MCTS	6	***8.42***	2.43	0.51	0.35	0.95	–
MCTS-50	9	15.16	4.16	0.86	0.22	1	3.01×10^{-11}
Random map1							
SL-MCTS	7	***9.55***	2.45	0.56	0.34	0.78	–
MCTS-50	16	26.60	12.85	0.78	0.22	1	0.017
Random map2							
SL-MCTS	7	***10.45***	2.36	0.68	0.144	0.83	–
MCTS-50	8	13.8	3.42	0.94	0.27	1	1.69×10^{-10}

The best value is highlighted in italic bold.

TABLE 4 Results of SL-MCTS and MCTS on different 16 × 16 maps.

	Best	Average length	SD-L	Average time	SD-T	Success rate(%)	p-value
Prior map							
SL-MCTS	12	***20.16***	13.61	7.12	5.19	0.74	–
MCTS-50	20	33.84	9.09	15.17	4.03	1	5.79×10^{-8}
Random map							
SL-MCTS	15	32.5	15.63	5.11	7.06	0.68	**0.07**
MCTS-50	15	***32.20***	15.05	12.72	6.04	1	—

The best value is highlighted in italic bold.

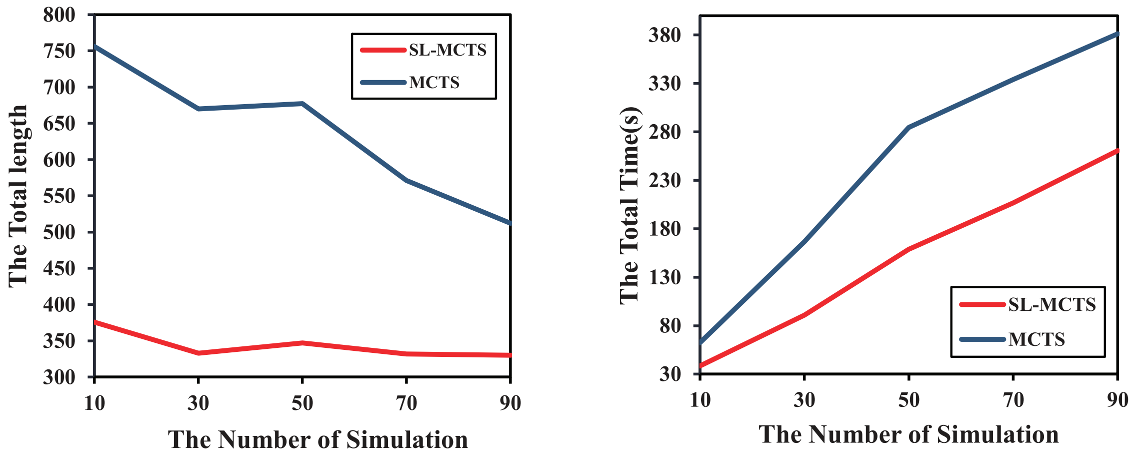


FIGURE 12
Comparison of the variations in total path length and total time consumption of SL-MCTS and MCTS with different numbers of simulations.

efficient in finding higher quality solutions than the traditional MCTS algorithms.

3.3.5. Test on dynamic environmental map

Finally, we tested the performance of SL-MCTS in a dynamic obstacle environment to deal with stochastic environments. In addition to the eight actions shown in Figure 1B, the robot's actions included the "wait" action. As shown in Figure 13, there was a dynamic obstacle in the environmental map, which is clockwise, and its movement trajectory was shown as an orange line. The trajectory has the starting point of (1, 2) and four turning points at (4, 2), (4, 4), (0, 3), and (0, 2). The robot's initial position was (0, 0) and the endpoint was (5, 5). Figure 13 shows two trajectories

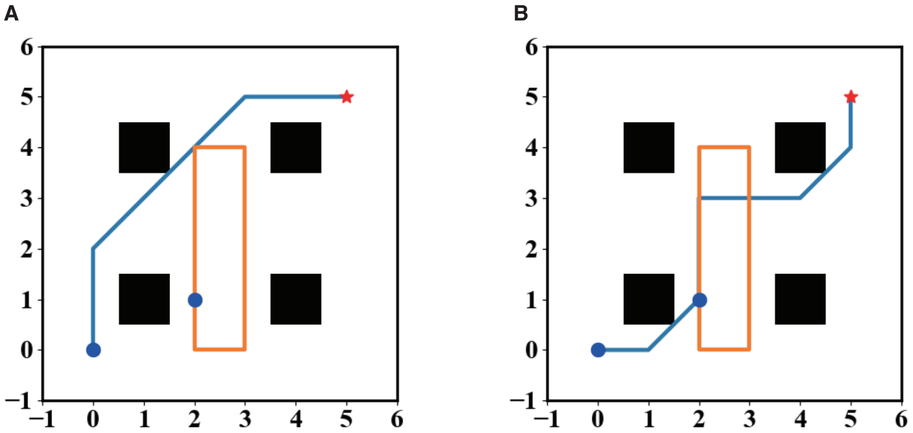


FIGURE 13
(A, B) The path planning results of the robot in a dynamic environment.

of the robot to deal with this dynamic obstacle. Figure 13A shows the robot successfully reached the destination without colliding with the dynamic obstacle. The robot chooses to bypass the area of the dynamic obstacle to reach the endpoint. Figure 13B shows the trajectory of the robot colliding with the dynamic obstacle at position (3, 3). To avoid collision and task failure, the robot waits in position (3, 2). These experiments demonstrated that SL-MCTS can handle dynamic environments. More related animations have been uploaded to the public repository (Liu, 2023).

4. Conclusion

Inspired by the idea of "self-player" for two-player zero-sum games, this paper proposes a self-learning single-player MCTS, named SL-MCTS, to continually enhance the problem-solving ability of agents in single-player scenarios. The main contributions of this paper include constructing the self-learning framework for single-player scenarios and designing an efficient evaluation method to assess the quality of the agent's strategies in the learning process. In the experiment section of this paper, a widely-renowned robot path planning scenario was utilized to validate the efficacy of SL-MCTS. In the self-learning process, the increasing Elo ratings of SL-MCTS show that the "self-learning" method for the single-player task is effective. The performance of SL-MCTS is also compared with that of MCTS, SP-MCTS, SP-MCTS-CRIPPA, and the currently popular collective intelligence algorithms in many different tasks. The results demonstrate that SL-MCTS can find better solutions with fewer iterations than other iteration-based algorithms, which indicates the convergence speed of SL-MCTS is faster. Additionally, in terms of time consumption, the speed of SL-MCTS in solving problems is faster than other comparative algorithms. It can solve problems in less than one-third of the time required by other algorithms. These indicate that the guidance of the PV-Network greatly improves the search efficiency and the resulting quality of SL-MCTS in path planning tasks. Furthermore, we validated the adaptability of SL-MCTS in many new environmental maps. The results show that SL-MCTS can find solutions with better quality in half the time required by MCTS-50. This experiment demonstrates that the problem-solving ability of SL-MCTS is universal across different environmental maps. Finally, we validated SL-MCTS's adaptability in a dynamic environment. The experimental results show that it can successfully solve tasks in dynamically complex scenes. In conclusion, this paper demonstrates that the mechanism of "self-learning" can be applied in single-player scenarios. It provides a new way for the agent with learning capabilities to break through its ceiling of problem-solving ability. Comparative experiments have confirmed that SL-MCTS can alleviate the common issues of slow convergence, poor search quality and inefficient search in traditional MCTS algorithms, while also significantly improving search speed.

In the future, we will further explore applying self-learning with other collective intelligence algorithms. We will also try to extend self-learning to improve the performance of the robotic arms in the continuous action space of the path planning problem.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

WL provided the original motivation and idea. YL further developed and implemented the idea, conducted the experiments, and produced the initial manuscript. WL and YL engaged in a thorough discussion and revision of this manuscript. YM, KX, and JQ checked the results and provided writing advice for the manuscript. ZG was responsible for the resources and revision of the manuscript and provided financial support. All authors contributed to the article and approved the submitted version.

Funding

This research was partially supported by Shanghai Municipal Science and Technology Major Project (No. 2021SHZDZX0103), Scientific Research Development Center in Higher Education Institutions by the Ministry of Education, China (No. 2021ITA10013), Engineering Research Center for Intelligent Robotics, Ji Hua Laboratory, Foshan, China (No. X190011TB190), the Shanghai Engineering Research Center of AI and Robotics, and the Engineering Research Center of AI and Robotics, Ministry of Education, China.

Acknowledgments

We acknowledge the support of the School of Academy for Engineering and Technology in University of Fudan of China.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aggarwal, S., and Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. *Comput. Commun.* 149, 270–299. doi: 10.1016/j.comcom.2019.10.014

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.* 47, 235–256. doi: 10.1023/A:1013689704352

Cheng, X., Li, J., Zheng, C., Zhang, J., and Zhao, M. (2021). An improved pso-gwo algorithm with chaos and adaptive inertial weight for robot path planning. *Front. Neurorobot.* 15, 770361. doi: 10.3389/fnbot.2021.770361

Coulom, R. (2008). "Whole-history rating: a Bayesian rating system for players of time-varying strength," in *International Conference on Computers and Games* (Berlin: Springer), 113–124. doi: 10.1007/978-3-540-87608-3_11

Crippa, M., Lanzi, P. L., and Marocchi, F. (2022). An analysis of single-player Monte Carlo tree search performance in sokoban. *Expert Syst. Appl.* 192, 116224. doi: 10.1016/j.eswa.2021.116224

Dai, X., Long, S., Zhang, Z., and Gong, D. (2019). Mobile robot path planning based on ant colony algorithm with a* heuristic method. *Front. Neurorobot.* 13, 1–15. doi: 10.3389/fnbot.2019.00015

Dam, T., Chalvatzaki, G., Peters, J., and Pajarinen, J. (2022). Monte-carlo robot path planning. *IEEE Robot. Autom. Lett.* 7, 11213–11220. doi: 10.1109/LRA.2022.3199674

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE Comput. Intell. Mag.* 1, 28–39. doi: 10.1109/MCI.2006.329691

Gelly, S., Kocsis, L., Schoenauer, M., Sebag, M., Silver, D., Szepesvári, C., et al. (2012). The grand challenge of computer go: Monte Carlo tree search and extensions. *Commun. ACM* 55, 106–113. doi: 10.1145/2093548.2093574

Gelly, S., and Wang, Y. (2006). "Exploration exploitation in go: UCT for Monte-Carlo go," in *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*. Vancouver, BC: MIT Press.

Halder, R. K. (2021). "Particle swarm optimization in global path planning for swarm of robots," in *Applying Particle Swarm Optimization*, ed R. K. Halder (Berlin: Springer), 209–232. doi: 10.1007/978-3-030-70281-6_12

Huang, J., Tan, Q., Li, H., Li, A., and Huang, L. (2022). Monte Carlo tree search for dynamic bike repositioning in bike-sharing systems. *Appl. Intell.* 52, 4610–4625. doi: 10.1007/s10489-021-02586-x

Kocsis, L. Szepesvári, C. (2006). "Bandit based monte-carlo planning," in *European Conference on Machine Learning* (Berlin: Springer), 282–293. doi: 10.1007/11871842_29

Kung, H.-L., Yang, S.-J., and Huang, K.-C. (2022). An improved Monte Carlo tree search approach to workflow scheduling. *Conn. Sci.* 34, 1221–1251. doi: 10.1080/09540091.2022.2052265

Lee, J., and Kim, D.-W. (2016). An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. *Inf. Sci.* 332, 1–18. doi: 10.1016/j.ins.2015.11.004

Li, J., Tinka, A., Kiesel, S., Durham, J. W., Kumar, T. S., Koenig, S., et al. (2021). Lifelong multi-agent path finding in large-scale warehouses. *Proc. AAAI Conf. Artif. Intell.* 35, 11272–11281. doi: 10.1609/aaai.v35i13.17344

Liu, Y. (2023). *Code of SL-MCTS*. Available online at: https://github.com/Liuyi61111/SL-MCTS (accessed May 24, 2023).

Luo, Q., Wang, H., Zheng, Y., and He, J. (2020). Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* 32, 1555–1566. doi: 10.1007/s00521-019-04172-2

Nielsen, F. (2020). On a generalization of the jensen-shannon divergence and the jensen-shannon centroid. *Entropy* 22, 221. doi: 10.3390/e22020221

Pellier, D., and Bouzy, B. Métivier, M. (2010). "An UCT approach for anytime agent-based planning," in *Advances in Practical Applications of Agents and Multiagent Systems: 8th International Conference on Practical Applications of Agents and Multiagent Systems (PAAMS 2010)* (Berlin: Springer), 211–220. doi: 10.1007/978-3-642-12384-9_26

Perez, D., Rohlfshagen, P., and Lucas, S. M. (2012). "Monte-carlo tree search for the physical travelling salesman problem," in *European Conference on the Applications of Evolutionary Computation* (Berlin: Springer), 255–264. doi: 10.1007/978-3-642-29178-4_26

Qi, X., Gan, Z., Liu, C., Xu, Z., Zhang, X., Li, W., et al. (2021). Collective intelligence evolution using ant colony optimization and neural networks. *Neural Comput. Appl.* 33, 12721–12735. doi: 10.1007/s00521-021-05918-7

Qi, X., Liu, C., Fu, C., and Gan, Z. (2018). Theory of collective intelligence evolution and its applications in intelligent robots. *Strateg. Study Chin. Acad. Eng.* 20, 101–111. doi: 10.15302/J-SSCAE-2018.04.017

Scariot, P., Manchado-Gobatto, F., Beck, W., Papoti, M., Ginkel, V., Gobatto, C., et al. (2022). Monocarboxylate transporters (MCTS) in skeletal muscle and hypothalamus of less or more physically active mice exposed to aerobic training. *Life Sci. 307*, 120872. doi: 10.1016/j.lfs.2022.120872

Schadd, M. P., Winands, M. H., Tak, M. J., and Uiterwijk, J. W. (2012). Single-player monte-carlo tree search for samegame. *Knowl. Based Syst.* 34, 3–11. doi: 10.1016/j.knosys.2011.08.008

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*. [preprint]. doi: 10.48550/arXiv.1707.06347

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270

Sturtevant, N. R. (2008). "An analysis of UCT in multi-player games," in *Computers and Games*, eds H. J. van den Herik, X. Xu, Z. Ma, and M. H. M. Winands (Berlin: Springer), 37–49. doi: 10.1007/978-3-540-87608-3_4

Xiong, N., Zhou, X., Yang, X., Xiang, Y., and Ma, J. (2021). Mobile robot path planning based on time taboo ant colony optimization in dynamic environment. *Front. Neurorobot.* 15, 642733. doi: 10.3389/fnbot.2021.642733

Yonetani, R., Taniai, T., Barekatain, M., Nishimura, M., and Kanezaki, A. (2021). "Path planning using neural a* search," in *International Conference on Machine Learning*, 12029–12039. Sydney, NSW: JMLR.org.

Yu, Z., Si, Z., Li, X., Wang, D., and Song, H. (2022). A novel hybrid particle swarm optimization algorithm for path planning of UAVS. *IEEE Internet Things J.* 9, 22547–22558. doi: 10.1109/JIOT.2022.3182798

Zhang, J., Xia, Y., and Shen, G. (2019). A novel learning-based global path planning algorithm for planetary rovers. *Neurocomputing* 361, 69–76. doi: 10.1016/j.neucom.2019.05.075

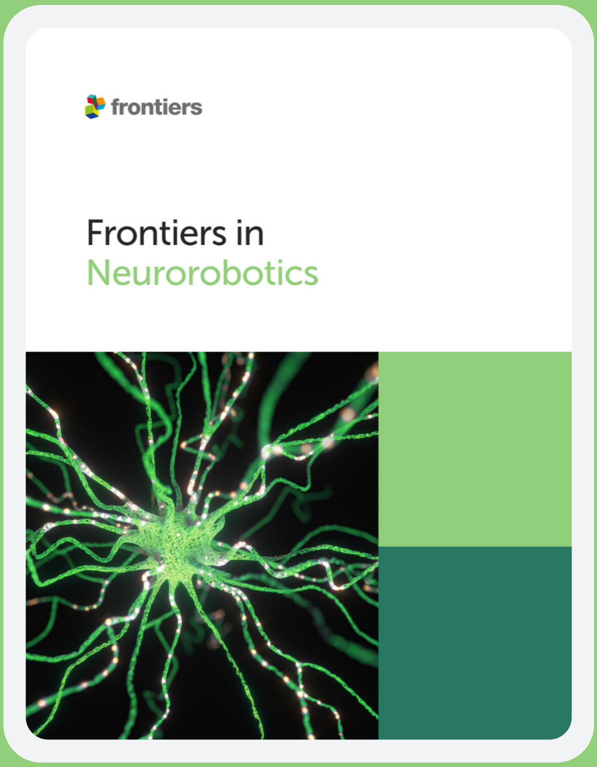
Frontiers in Neurorobotics

Investigates embodied autonomous neural systems and their impact on our lives

Part of the most cited neuroscience series, this journal advances understanding of neurorobotics - from prosthetic devices to brain machine interfaces, and wearable systems to home appliances.

Discover the latest Research Topics

See more →



frontiers

Frontiers in Neurorobotics

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact