International Journal of Advanced AI Applications

Volume2,Issue2,Feb.2026

Online ISSN 3104-9338
Print ISSN 3104-932X

Hong Kong Dawn Clarity Press Limited
http://www.dawnclarity.press/index.php/ijaaa

TABLE OF CONTENTS

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Design of an Adaptive Stair-Climbing Robot Based on Heterogeneous Dual-Core Intelligent Control Technology

Jialing Tang*, Xiaoying He

Department of Intelligent Manufacturing Engineering, Chengdu College of University of Electronic Science and Technology of China, Sichuan, Chengdu, 611731, China

Received: December 26, 2025

Revised: January 6, 2026

Accepted:January, 2026

Published online: January, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 2 (February 2026)

* Corresponding Author: Jialing Tang (2025424068@qq.com)

Abstract. With the deepening trend of societal aging, the demand for mobile robots in scenarios such as elderly assistance, disability aid, logistics, and rescue is growing. Navigating stairs in complex, unstructured environments has become a key challenge in robotics. Traditional wheeled, tracked, or legged robots suffer from weak adaptability, insufficient stability, or high cost. This paper designs an adaptive stair-climbing robot utilizing a heterogeneous dual-core control architecture built with an STM32H743 microcontroller and a Raspberry Pi 4B. It integrates multiple sensors including an RGB-D camera, an Inertial Measurement Unit (IMU), and encoders. The Raspberry Pi 4B serves as the upper-layer intelligent decision-making core, performing planning and decision-making through fuzzy logic and Model Predictive Control (MPC). The STM32H743 acts as the lower-layer real-time control core, achieving precise execution via PID control. The robot can adapt to stairs with slopes of 30°–45° and step heights of 150–200 mm made of different materials, maintaining a stability margin of no less than 20 mm during climbing. Compared to traditional tracked robots, the stability margin is improved by over 35%. The robot demonstrates good stability and robustness in various stair environments, providing an innovative technical approach for mobile robots in complex terrains.

Keywords: *Adaptive Stair-climbing Robot; Heterogeneous Dual-core Control; Multi-Sensor Fusion; PID Control*

1. Introduction

To address stair terrain, related research domestically and internationally has primarily focused on three categories of robots: wheeled, tracked, and legged. Wheeled mechanisms offer high efficiency but poor obstacle-crossing capability. Tracked mechanisms improve possibility

to some extent but often lack stability during stair ascent, being prone to posture instability. Legged robots have the best environmental adaptability but are limited by complex control logic and high manufacturing costs [1]. The core performance differences among different types of mobile mechanisms are shown in Table 1. In recent years, the use of hybrid mobile mechanisms has become a research focus for such compromise solutions [3]. Although existing hybrid mechanisms balance movement efficiency and obstacle-crossing ability, most rely on pre-programmed gaits. In unknown stair environments, they exhibit algorithmic lag in dynamic center-of-gravity adjustment, with response delays commonly exceeding 80 ms. In contrast, a heterogeneous dual-core architecture can compress decision-making delays to within 50 ms. The multi-wheel-group mechanism combines the efficiency of wheeled systems with the obstacle-crossing capability of tracked systems, allowing flexible switching between wheeled and tracked modes, providing a solid mechanical foundation for adapting to stair terrain.

Most existing research focuses on mechanical structure improvements or relies on fixed gaits preset with stair parameters. When dealing with unknown or variable-parameter stair environments, the "perception-decision-adaptation" intelligent control capability of such solutions remains insufficient. Embedding an intelligent system into a multi-wheel-group mobile platform to endow it with autonomous adaptation capability is key to solving the problem.

Table 1. Performance comparison of different mobile mechanisms for stair climbing.

Mobile Mechanism Type	Movement Efficiency	Obstacle-Crossing Capability	Stair-Climbing Stability
Wheeled Mechanism	High	Weak	Poor (Prone to Slipping)
Tracked Mechanism	Medium	Medium	Fairly Poor (Prone to Instability)
Legged Mechanism	Low	Strong	Good
Hybrid Mechanism	Medium	Medium	Average
Adaptive Multi-Wheel-Group Mechanism	Medium-High	Strong	Excellent

This paper proposes an innovative "heterogeneous dual-core intelligent control + multi-sensor fusion" solution, developing an adaptive stair-climbing robot. The heterogeneous dual-core architecture balances real-time control and intelligent decision-making. The upper layer uses a fuzzy control algorithm, which does not rely on an accurate mathematical model, to achieve dynamic decision-making. The lower layer uses PID control for accurate execution. The aim is to endow the robot with autonomous adaptation capability in unknown stair environments, overcoming the limitations of traditional solutions, and providing a new approach for autonomous robot navigation in unstructured environments.

2. Overall Design of the Adaptive Stair-Climbing Robot

The adaptive stair-climbing robot system is a complex system integrating mechanics, electronics, control, and information processing. Its overall design follows the principles of modularity, intelligence, and high reliability. The entire system consists of three core modules: the mechanical body module, the sensing and actuation module, and the heterogeneous dual-core intelligent control module. The overall framework diagram is shown in Figure 1, illustrating the information and control flow from environmental perception to motion execution.

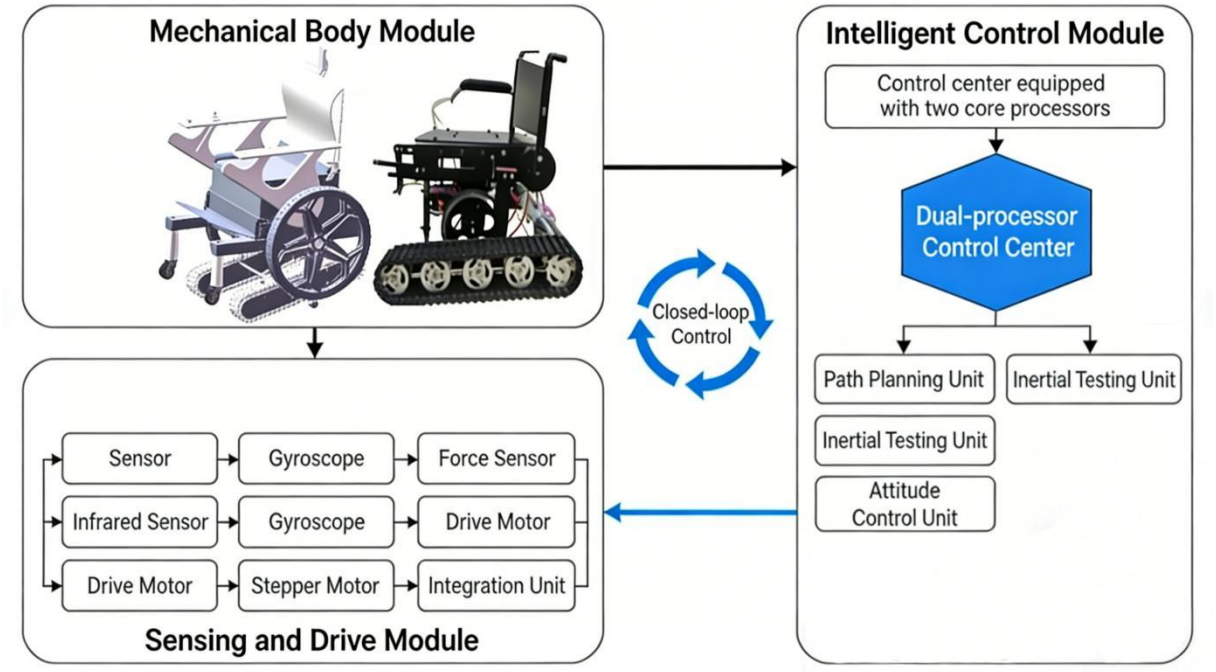

Figure 1. Overall system framework diagram.

2.1. Mechanical Body Module

The mechanical body is the physical carrier of the robot, as shown in Figure 2.

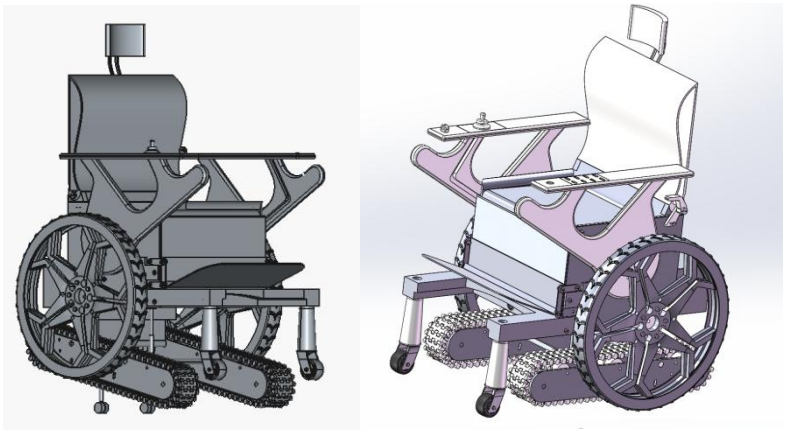

Figure 2. Rendering of the adaptive stair-climbing robot.

It adopts a multi-wheel-group mobile mechanism that combines the efficiency of wheeled systems with the obstacle-crossing ability of tracked systems. The mechanism consists of 4 symmetrically distributed wheel groups. Each wheel group integrates a driving wheel, auxiliary support wheels, and an elastic tensioning component. Based on feedback from step contact, it can automatically adjust the support angle and tension of the wheel groups, ensuring multiple support points provide stable support force during climbing. This retains the high movement efficiency of wheeled mechanisms while possessing the strong obstacle-crossing capability of tracked mechanisms, effectively preventing tipping over [4].

2.2. Sensing and Actuation Module

The sensing and actuation module is the "nerves" and "muscles" for the robot to perceive the environment and execute actions. It includes a depth vision sensor (e.g., RGB-D camera) for detecting the distance, angle, and step height of stairs ahead; an Inertial Measurement Unit (IMU) for measuring changes in the robot's own posture; encoders for feeding back the actual positions of joints; and DC servo motors or steering gears as power outputs. The core perception task, undertaken by the RGB-D camera for stair environment detection, requires accurate identification and parameter extraction of stair targets. The complete logical flow for stair target detection is shown in Figure 3. This process takes color images and depth point cloud data as input, achieves stair contour segmentation and key parameter fitting through multi-stage processing, and obtains reliable environmental perception data. Based on this, the robot makes adaptive stair-climbing decisions [5].

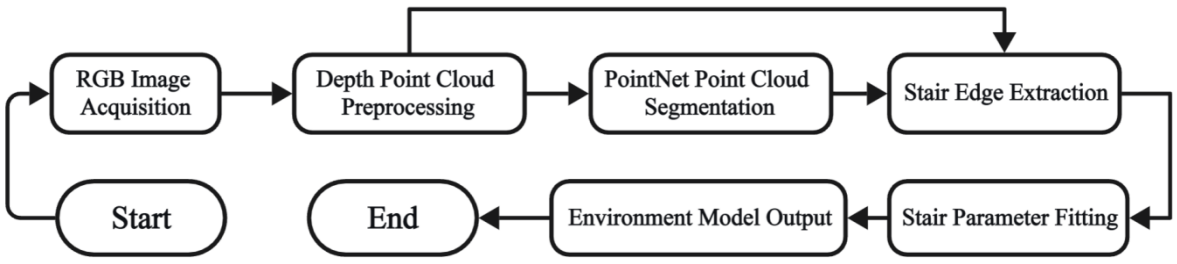

Figure 3. Flowchart of stair target detection.

2.3. Heterogeneous Dual-Core Intelligent Control Module

The heterogeneous dual-core intelligent control module is the intelligent core of the entire robot, adopting a dual-processor structure with different architectures. The implementation flow is shown in Figure 4. The STM32H743 microcontroller serves as the real-time control core, running the RT-Thread operating system. Its main functions are time-sensitive basic

operations, millisecond-level motor servo control, and rapid collection and filtering of multi-channel sensor signals. In contrast, the more powerful Raspberry Pi 4B, running the ROS (Robot Operating System), handles computationally intensive intelligent decision-making tasks such as environmental recognition, multi-sensor data fusion, and real-time motion planning. The two cores communicate in real-time via a high-speed serial interface (UART), continuously exchanging control commands and system states, forming a perfect closed-loop autonomous control cycle from perception to decision-making and execution, endowing the robot with both rapid reflex capability and complex reasoning ability.

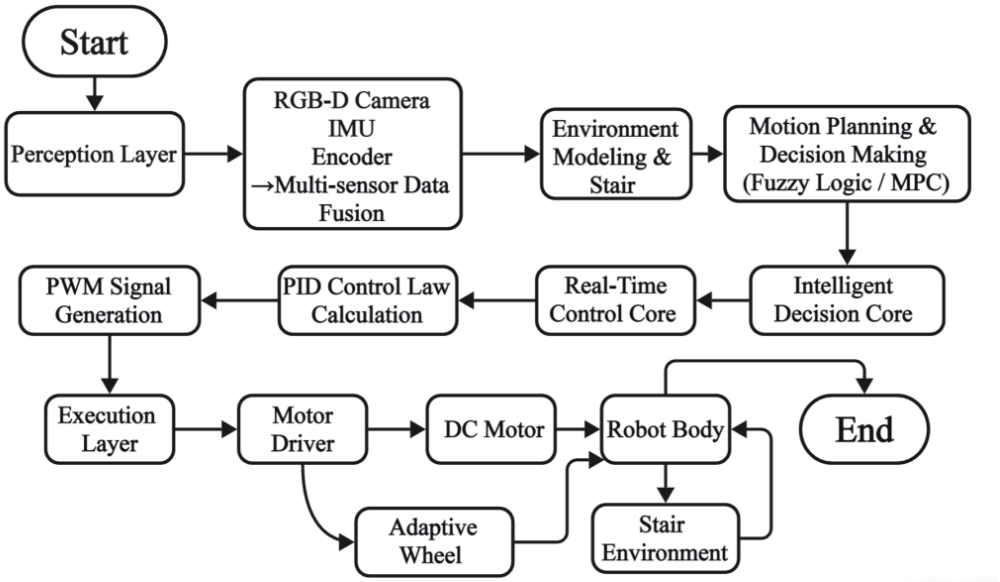

Figure 4. Flowchart of the heterogeneous dual-core intelligent control module.

3. Robot Mechanical Structure and Motion Mechanism

3.1. Adaptive Walking Mechanism

The multi-wheel-group mobile mechanism, which fuses wheeled efficiency and tracked obstacle-crossing ability, is the foundation for realizing the robot's stable stair-climbing function. This mechanism abandons the structural limitations of traditional single wheeled or tracked designs. It employs 4 independently driven wheel group units arranged in a rectangular array on both sides of the body. Each wheel group unit includes an 80 mm diameter polyurethane driving wheel, auxiliary support wheels, and an elastic tensioning link rod with a stroke of 0-120 mm. A torque sensor (model: TJH-803) at the wheel group pivot triggers wheel group posture adjustment.

When a wheel group contacts the vertical face of a step and the pressure exceeds a set threshold of 5 N, the equivalent motor (JGA25-370) activates, actively lifting the wheel group

to form a stable temporary auxiliary support point. The other wheels continue moving smoothly to push the body forward. After the wheel group completely crosses the vertical face and lands on the step tread, the tensioning link automatically resets. Through this physical interaction-based feedback and independent switching, the robot achieves dynamic adaptation. Without relying on complex external sensor systems, it balances the efficiency of wheeled mechanisms and the multi-support-point obstacle-crossing capability of tracked systems through natural interaction between wheel groups and steps, demonstrating flexibility in adapting to steps of varying heights and slopes.

3.2. Kinematics and Stability Analysis

To quantitatively analyze the robot's motion, a simplified kinematic model was established. Let the projection of the robot's center of gravity on the horizontal plane be $G(x_g, y_g)$, and the contact points of each wheel group with the ground be $P_i(x_i, y_i),\ i = 1,2,3,4$. During stair climbing, the robot's static stability margin SM can be defined as the minimum value of the shortest distances from the center of gravity G to each side of the current support polygon.

$$SM = imindistance\big(G, edge\ i(Polygon)\big)$$

The robot is statically stable only when $SM > 0$. In dynamic processes, dynamic stability must be evaluated by calculating the Zero Moment Point (ZMP) or the rate of tilt angle change, combined with IMU data. The wheel group alternating support strategy designed in this paper aims to actively maintain a large support polygon, keeping SM above a safe threshold throughout the climbing process.

To achieve precise tracking of the preset trajectory and accurate control of the motor driving torque, it is necessary to establish the system's kinematic and dynamic models. In kinematics, the D-H parameter method is used to establish coordinate systems, with the base at the body center and links at each wheel group joint. Deriving the forward kinematics equation relates joint variables such as wheel group speed and tensioning angle to the robot's overall pose (position, orientation), providing the basis for the inverse solution in multi-wheel-group coordinated trajectory planning. In dynamics, a system dynamic model is constructed based on the Lagrange equation, focusing on analyzing the force balance relationships during different phases such as wheel group contact with steps and lifting for obstacle crossing. This includes the robot's own gravity, inertial forces generated by motion, ground contact reaction forces, and motor driving torques, estimating the peak torque requirements for each joint. This provides theoretical support for motor selection and parameter tuning of the underlying PID controller.

In actual control, simplified forms of these models are used for real-time prediction and planning by the upper-layer Raspberry Pi decision core.

3.3. Obstacle-Crossing Stability and Posture Adjustment

During stair climbing, stability is the primary condition for ensuring task success and the robot's own safety. The special stair environment causes the robot's center of gravity position to continuously change with climbing height, which can easily lead to forward/backward or lateral tipping. Therefore, obstacle-crossing stability analysis and active posture adjustment strategies are key links in the overall design. The robot's posture adjustment strategy is shown in Figure 5, achieving stable climbing by dynamically adjusting support point positions. The robot's stability is quantitatively assessed by calculating the position of the center of gravity within the support polygon; this assessment metric is the static stability margin. For dynamic processes like climbing, professional concepts such as the Zero Moment Point (ZMP) must also be considered [8]. The core feature of the adaptive walking mechanism designed in this paper is multi-point alternating support, which actively maintains a large stable support area. Cooperating with the heterogeneous dual-core control system, the intelligent decision core solves relevant data in real-time, including body tilt angle and angular velocity information from the IMU, and support leg position information from joint encoders. Based on this data, it dynamically calculates the robot's real-time center of gravity and stability margin. The Raspberry Pi 4B then generates motion trajectories and posture compensation commands according to the calculation results and sends them to the real-time control core via the UART asynchronous serial port. The real-time control core communicates with peripherals like the IMU and encoders using the SPI interface, effectively ensuring high-speed acquisition of underlying sensor data. The real-time control core utilizes its high timer resolution to execute high-speed PID control algorithms, converting received commands into precise PWM driving signals for each joint motor, ultimately achieving motion tracking and dynamic stability.

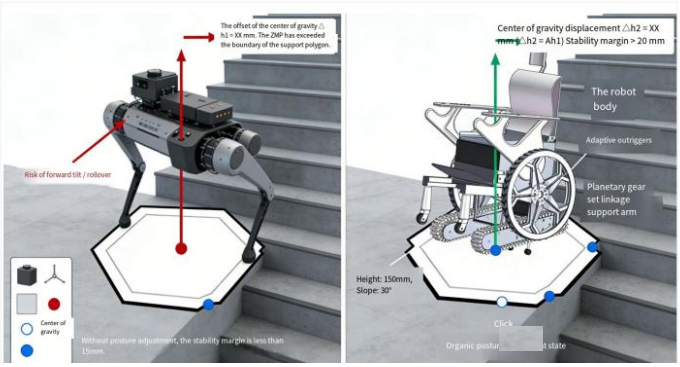

Figure 5. Posture adjustment strategy comparison diagram.

4. Design of the Heterogeneous Dual-Core Intelligent Control System

4.1. Heterogeneous Dual-Core Hardware Architecture and Task Allocation

The hardware architecture of the dual-core intelligent control system is the foundational platform for achieving high-performance control. The specific hardware configuration, task division, and key performance parameters are shown in Table 2.

Table 2. Hardware task division of the heterogeneous dual-core control system.

Core Type	Processor Model	Operating System	Response Time	Interface Connected To
Real-Time Control Core	STM32H743 (ARM Cortex-M7)	RT-Thread	Microsecond level (≤10 μs)	Motor drivers, joint encoders, IMU
Intelligent Decision Core	Raspberry Pi 4B (ARM Cortex-A72)	Linux + ROS Noetic	Millisecond level (≤50 ms)	RGB-D camera, Real-Time Control Core

This design adopts a heterogeneous dual-processor solution, with the real-time control core focusing on "fast response and precise execution" and the intelligent decision core focusing on "complex data processing and dynamic decision-making." Figure 6 visually presents the collaborative hardware foundation of the heterogeneous dual cores.

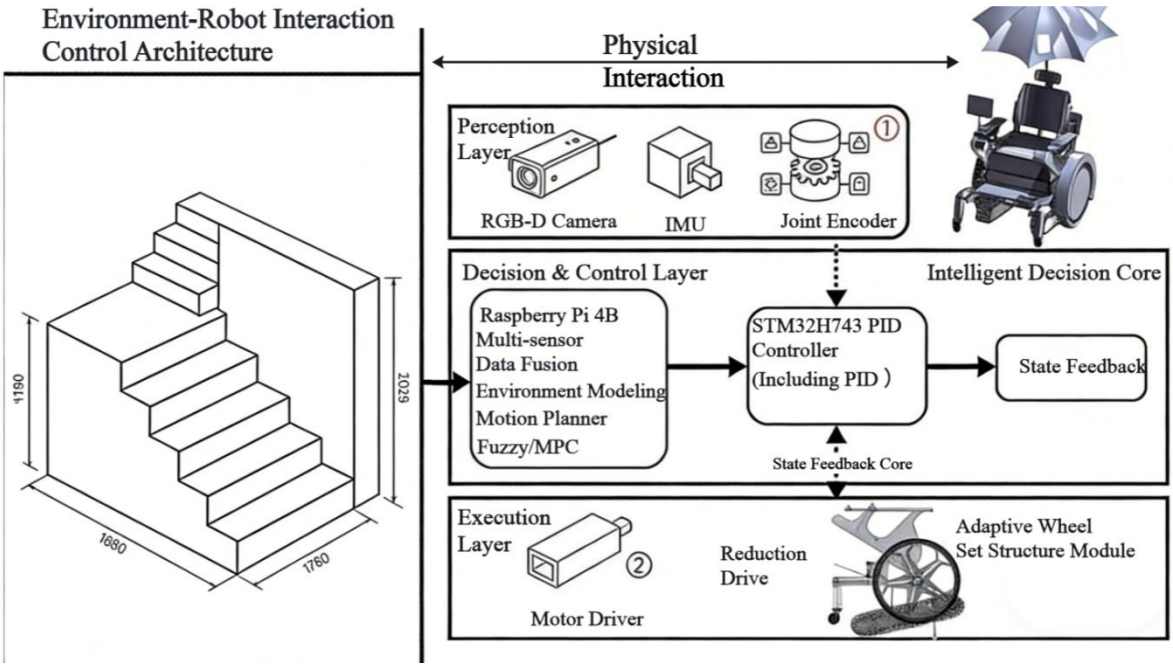

Figure 6. Hardware architecture and task allocation diagram.

The real-time control core uses an STM32H7 series microcontroller with an ARM Cortex-M7 core. Its specific pin assignment and hardware connection design are shown in Figure 7. Its maximum main frequency can reach 480 MHz, and it possesses abundant timer/PWM output channels and nanosecond-level interrupt response capability, fully meeting the stringent "low

latency, high precision" requirements of underlying control. The core connects directly to hardware devices such as hub motor drivers, wheel pair encoders, and the IMU via high-speed General Purpose Input/Output (GPIO) and Serial Peripheral Interface (SPI). It runs underlying programs on the lightweight real-time operating system (RT-Thread) to receive speed/position feedback signals from wheel pair encoders, complete the closed-loop PID control of hub motors ensuring precise tracking of multi-wheel-group motion trajectories; and perform filtering preprocessing on the raw three-axis acceleration and angular velocity data collected by the IMU to reduce noise interference. It parses target posture commands and wheel group power distribution commands issued by the intelligent decision core, converting them into specific driving PWM signals to achieve coordinated motion control of each wheel group motor.

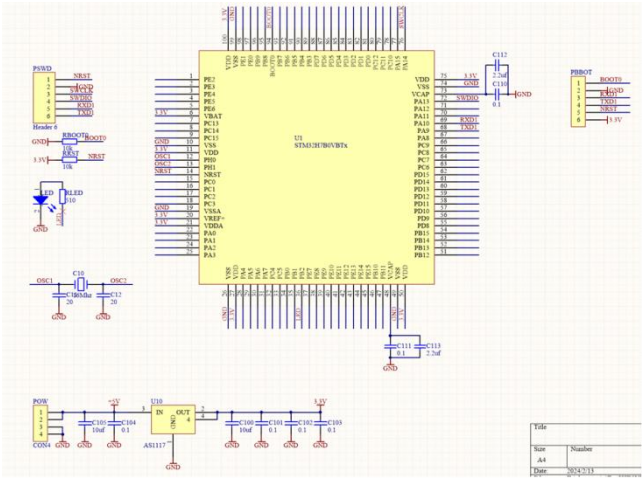

Figure 7. Hardware diagram of STM32H743 core pins.

The intelligent decision core is selected from the ARM Cortex-A series application processors, specifically the Raspberry Pi 4B. Its quad-core Cortex-A72 processor offers strong computing performance, capable of running the Linux operating system and ROS Noetic (Robot Operating System) smoothly, providing hardware support for deploying complex algorithms. The core is responsible for non-real-time but highly complex intelligent processing tasks. It receives color images and depth data from the RGB-D camera via a USB 3.0 interface, then fuses preprocessed IMU data and encoder feedback on actuator status information. Using multi-sensor data fusion algorithms, it constructs a "robot-environment" combined state model. The system can identify key environmental parameters such as stair slope, step height/depth, generate adaptive multi-wheel-group coordinated motion trajectories for the current environment, and immediately generate compensation commands if posture deviation risks are detected, sending them to the real-time control core.

The two cores achieve bidirectional communication via a high-speed asynchronous serial

port (UART) at a data transmission rate of 115200 bps. This communication method forms a classic closed-loop solution for heterogeneous dual-core robots, enabling a 10 μs state data upload and a 50 ms command issuance response [4]. The real-time control core uploads state information such as motor speed, body tilt angle, and trajectory position collected by sensors every 10 μs, providing a dynamic data foundation for intelligent decision-making. The intelligent decision core issues updated motion and posture correction commands every 50 ms, which the real-time control core quickly responds to and executes.

This separation of responsibilities avoids potential performance conflicts between real-time control and complex decision-making tasks within a single processor. Efficient communication achieves overall coordinated control, providing stable hardware support for the robot's stair climbing and environmental adaptation.

4.2. Sensor Data Fusion and Environmental Modeling

Accurate perception of the environment is a prerequisite for the robot's autonomous adaptive climbing. The multi-modal sensors on the robot provide complementary environmental information. The RGB-D camera acquires color images and depth information from the environment in front of the robot [10]. With the help of point cloud processing algorithms, stair surfaces can be segmented, and stair step heights and depths can be extracted to build a geometric model of the stairs ahead. The IMU provides the robot's body three-axis acceleration and three-axis angular velocity. Through attitude calculation algorithms (such as complementary filtering or Kalman filtering), the robot's pitch and roll angles relative to the direction of gravity can be estimated in real-time, which are key parameters for assessing body posture stability. In complementary filtering, the low-pass filter cutoff frequency for IMU accelerometer data is set to 5 Hz, and the high-pass filter cutoff frequency for gyroscope data is set to 0.5 Hz. By fusing attitude data with a weighting coefficient k=0.98, noise interference on tilt detection is effectively reduced. Joint encoders accurately feedback the rotation angle or extension length of each adaptive leg. Combined with the robot's kinematic model, the pose of the robot chassis relative to support points can be derived.

The data fusion center on the Raspberry Pi 4B (intelligent decision core) deeply fuses visual data collected from the local RGB-D camera with the IMU attitude data and detailed encoder information preprocessed and acquired in real-time by the real-time control core (STM32H743) via the SPI bus [5], constructing an integrated robot-environment state model. The model flowchart is shown in Figure 8.

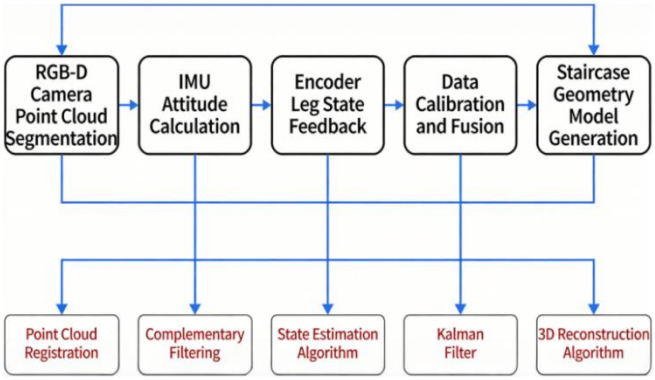

Figure 8. Flowchart of multi-sensor data fusion.

4.3. Adaptive Motion Planning and Stability Control Algorithm

Based on the integrated environmental state model, the core algorithms for adaptive motion planning and stability control operate. Motion planning generates a reference path for safely traversing all steps from the current position according to identified stair parameters (slope, step height/depth) and the robot's kinetic constraints, planning differentiated motion sequences for multiple wheel groups. As shown in Figure 9, to accurately convert the reference trajectory into motor operation commands, the system adopts a dual closed-loop control strategy.

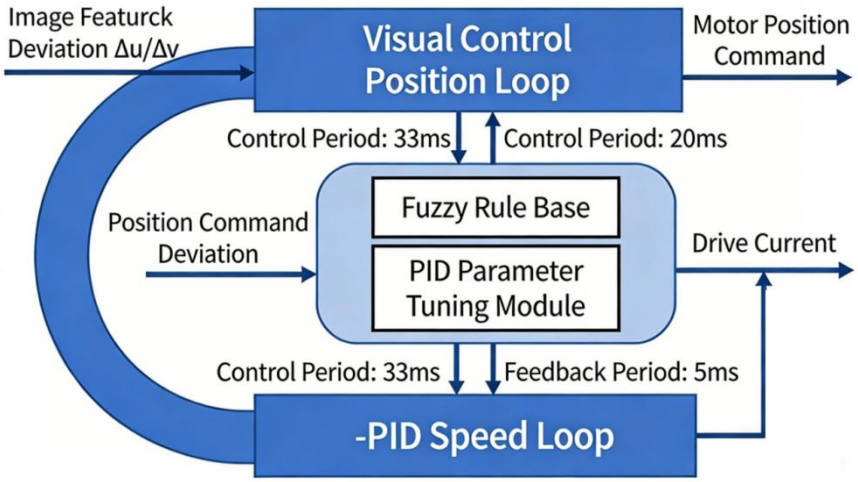

Figure 9. Flowchart of the dual closed-loop control.

The trajectory planner determines the movement path (position and orientation) of the robot base and the motion sequence for adjusting each leg (when to lift, lower, and anticipate) based on the area [11]. Because the stair environment may have uncertainties, the planner needs online re-planning capability to cope with updated situations or emergencies. The stability control algorithm works closely with the planner, operating as a supervisory and compensation layer. It continuously monitors the real-time stability margin calculated from IMU data and the kinematic model, specifically as shown in Figure 10. The logic between the upper and lower

layers of the algorithm is annotated, indicating command interaction and compensation mechanisms between the two layers, reflecting the layered collaboration of the algorithm.

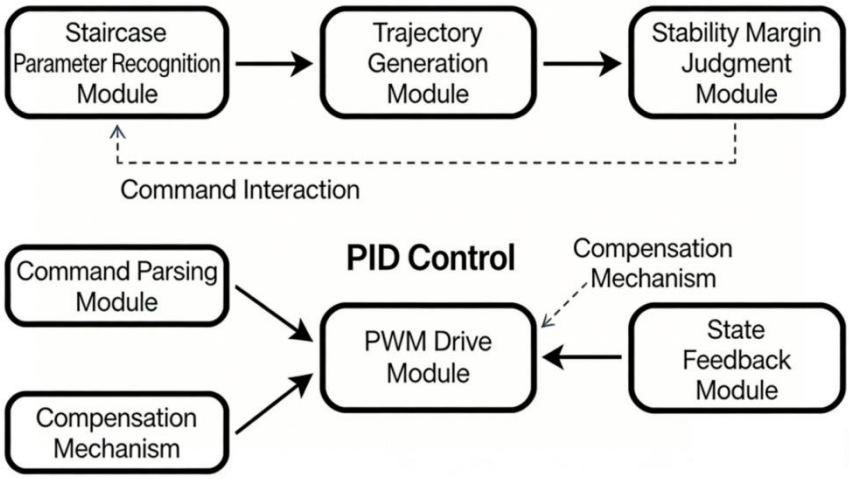

Figure 10. Closed-loop diagram of adaptive motion planning and stability control algorithm.

4.4. Experimental Verification

To verify the performance of the adaptive stair-climbing robot designed in this paper in complex stair environments, a multi-scenario test platform was built. Tests selected common stair types in civil buildings and simulated extreme scenarios such as wet, slippery, and irregular steps. Test indicators included average climbing speed, stability margin, and continuous stair-climbing success rate. Each group of tests was repeated 3 times, with the final result being the average value. The selection of test scenarios referenced the application requirements for elderly and disabled assistive robots. Specific test parameters and results are shown in Table 3.

In the standard stair scenario, the robot achieved a 100% success rate, maintaining a stability margin above 20 mm. Through coordinated adjustment of multiple wheel groups, fluctuations in body tilt angle were controlled within ±3°, demonstrating good stability performance. In complex scenarios such as irregular and slippery stairs, the robot's average ascent speed decreased, but the stability margin still met safety requirements, with a continuous climbing success rate of no less than 90%. Experimental results indicate that the mechanical structure and heterogeneous dual-core intelligent control strategy designed in this paper can effectively adapt to different types of stair environments, verifying the effectiveness and practicality of the proposed solution.

Table 3. Stair-climbing performance test results.

Test Stair Type	Step Height (mm)	Step Slope (°)	Surface Material	Average Climbing Speed (m/s)	Stability Margin (mm)	Climbing Success Rate (%)
Residential Standard Stairs	150	30	Concrete	0.22	28	100
Public Building Wide Stairs	180	35	Ceramic Tile	0.18	25	100
Worn Irregular Stairs	160 (±15)	32	Marble	0.15	22	97
Simulated Slippery Stairs	170	38	Floor Tile	0.13	20	95

5. Conclusion

This paper presents an adaptive stair-climbing robot based on heterogeneous dual-core intelligent control technology. It integrates a multi-wheel-group mechanical structure combining wheeled efficiency and tracked obstacle-crossing ability with a layered control system, addressing the pain point of poor adaptability of traditional robots in complex stair environments. The dual-core architecture balances real-time control and intelligent decision-making, ensuring environmental perception accuracy. Experiments have verified the stability and practicality of the solution, providing an innovative solution for the development of mobile robots in complex terrains.

Acknowledgement

This work was supported by Professor He Xiaoying. I would like to express my sincere gratitude to my supervisor, Professor He Xiaoying, for her critical guidance and financial support throughout this research. She provided me with valuable research direction, and during the writing and repeated revision of the thesis, her meticulous review and highly constructive feedback were essential in refining this paper. Her rigorous and pragmatic approach to research, combined with her kind and generous mentorship, has set an enduring example for me throughout my academic journey.

References

[1] Luo, W., Zhang, C., et al. (2025). 移动机器人路径规划算法体系综述与展望 [A Review and Future Outlook of Path Planning Algorithm Systems for Mobile Robots]. *Computer Engineering and Applications*, 1-18. https://link.cnki.net/urlid/11.2127.TP.20251230.1853.014.[in Chinese]

[2] Song, X., Zhang, Y., Dong, T., & Li, F. (2025). DHRN: A robot autonomous navigation

method in crowds based on heterogeneous framework. *Neurocomputing*, 132375.

[3] Shen, Z., Wang, R., Wang, L., Lu, W., & Wang, W. (2025). Application Research on General Technology for Safety Appraisal of Existing Buildings Based on Unmanned Aerial Vehicles and Stair-Climbing Robots. *Buildings*, 15(22), 4145.

[4] Shao, C. (2025). 基于 STM32H743 的无刷直流电机 FOC 自动化控制系统设计 [Design of an FOC Automatic Control System for Brushless DC Motors Based on STM32H743]. *Automation & Instrumentation*. (09), 73-77. https://doi.org/10.14016/j.cnki.1001-9227.2025.09.073.[in Chinese]

[5] Yang, D. (2024). 消防爬楼运输机器人研究与设计 [Research and Design of a Fire-Fighting Stair-Climbing Transport Robot]. *Modern Manufacturing Technology and Equipment*, 60(12), 84-86. https://doi.org/10.16107/j.cnki.mmte.2024.0840.[in Chinese]

[6] Zhao, J. (2024). 基于多传感器融合的地面移动机器人定位方法研究 [Research on Positioning Methods for Ground Mobile Robots Based on Multi-Sensor Fusion]. Chongqing University. https://doi.org/10.27670/d.cnki.gcqdu.2024.001574.[in Chinese]

[7] Zheng, P. (2023). 基于多传感器融合的移动机器人定位和路径规划方法研究 [Research on Positioning and Path Planning Methods for Mobile Robots Based on Multi-Sensor Fusion]. *Southeast University*. https://doi.org/10.27014/d.cnki.gdnau.2023.004282.[in Chinese]

[8] Xiao, Y. (2023). 可重构移动机器人多模态运动控制关键技术研究 [Research on Key Technologies of Multi-Modal Motion Control for Reconfigurable Mobile Robots]. *Chongqing University.* https://doi.org/10.27670/d.cnki.gcqdu.2023.002758.[in Chinese]

[9] Su, L., Zhang, L., Shao, J., et al. (2023). 楼梯清洁机器人的爬楼规划研究[Research on Stair-Climbing Path Planning for Stair-Cleaning Robots]. Journal of Jiangxi Normal University (Natural Science Edition), 47(1), 77-81. https://doi.org/10.16357/j.cnki.issn1000-5862.2023.01.10. [in Chinese]

[10] Zhang, J., Wang, Z., Zhang, Y. (2022). ADAMS 的电梯载荷测试机器人稳定性仿真 [Stability Simulation of Elevator Load Testing Robots Based on ADAMS]. *Mechanical Management and Development.* 37(07), 11-14. https://doi.org/10.16525/j.cnki.cn14-1134/th.2022.07.004.[in Chinese]

[11] Chen, Y., Li, Y. (2022). 驱动式多功能助老爬楼机设计 [Design of a Driven Multi-Functional Stair-Climbing Machine for Elderly Assistance]. Development & Innovation of Machinery & Electrical Products. 35(04), 54-56.

[12] Li, R., Xiao, Z., Chen, G. 基于曲柄摇杆机构的多足爬楼机器人设计 [Design of a Multi-Legged Stair-Climbing Robot Based on a Crank-Rocker Mechanism]. *Modern Manufacturing Technology and Equipment*. 58(07), 70-72. https://doi.org/10.16107/j.cnki.mmte.2022.0468.[in Chinese]

[13] Sun, S., Zhai, Z., Chen, L., et al. (2026). 基于 Arduino 的老人和儿童家庭服务机器人 [A Home Service Robot for the Elderly and Children Based on Arduino]. *Mechanical & Electrical Engineering Technology*. 1-8. https://link.cnki.net/urlid/44.1522.TH.20251217.1623.003.

[14] He, L., Chen, Y., Qi, J. (2025). 基于激光雷达-IMU 双层耦合的移动机器人位姿估计 [Mobile Robot Pose Estimation Based on LiDAR-IMU Double-Layer Coupling]. Computer Applications and Software. 42(12), 65-70.

[15] Duan, H. (2025). 基于多传感器数据融合的工业机器人运动路径规划研究 [Motion Path Planning for Industrial Robots Based on Multi-Sensor Data Fusion]. *Die & Mould Manufacture*. 25(12), 198-200. https://doi.org/10.13596/j.cnki.44-1542/th.2025.12.067.[in Chinese]

Online ISSN: 3104-9338
Print ISSN: 3104-932X

EchoKG: A Dynamic user Preference Knowledge Graph In-vehicle Dialogue System Based on Ebbinghaus Forgetting Curve

Yuqian Liang*

Chengdu University of Technology, China, 610059

Received: January 19, 2026

Accepted: January 21, 2026

Published online: January 21, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 2 (February 2026)

* Corresponding Author: Yuqian Liang (3083004993@qq.com)

Abstract. With the increasing integration of large language models (LLMs) into intelligent vehicle cockpits, achieving efficient, accurate, and personalized interactions with long-term memory capabilities has become a key challenge. Existing vector retrieval methods suffer from context inflation issues, while static knowledge graphs struggle to capture the time-varying nature of user preferences. This paper proposes the EchoKG framework, which for the first time mathematically models the Ebbinghaus forgetting curve as a dynamic weight mechanism for knowledge graph nodes, enabling the natural decay and reinforcement of user preferences. By introducing memory strength S and last access time, EchoKG dynamically manages the lifecycle of memories. Experimental results on the fully open-source dataset EchoCar-Public demonstrate that compared to MemoryBank, static knowledge graphs, and GPT-4o Memory, EchoKG reduces the average context length by 32%, increases the F1 score for intent recognition by 5.1%, and improves the personalized consistency score by 0.68 points, while maintaining a response latency within 800ms.

Keywords: *Large Language Model, Dialogue System, Knowledge Graph, Forgetting Curve.*

1. Introduction

Intelligent cockpits are evolving from the traditional "command-execution" mode to the "proactive - empathetic" intelligent companion mode. The ideal in-car assistant not only needs to understand the current driving instructions (such as "turn on the air conditioner"), but also needs to have the ability of Long-Term Memory that spans time periods. For instance, when a user sets the air conditioner to 26℃ several times in a row during winter, the system should automatically recommend this temperature in the following winter and "forget" this setting in summer. This long-term personalized service based on historical interaction is at the core of

enhancing user stickiness and in-cabin experience [1].

At present, memory enhancement schemes based on large language models (LLMS) mainly face two major challenges. The first is the vector memory dilation and retrieval noise phenomenon. Methods represented by MemoryBank convert historical dialogues into vector storage [2]. With the increase of usage time, the scale of the vector library grows exponentially, which not only leads to an increase in retrieval Latency, but also introduces a large amount of irrelevant historical noise, occupies the limited Context Window of the LLM, and even triggers "hallucinations". Secondly, there is the rigidity of static knowledge graphs. Although knowledge graphs (KGS) can provide structured fact storage, traditional KGS are static. Users' preferences are dynamic and fluid (for instance, a user might shift from preferring "rock" to "light music"). Static KG has difficulty eliminating outdated information through the "forgetting" mechanism, leading to recommendation conflicts.

In response to the above issues, inspired by cognitive psychology, this paper proposes the EchoKG framework. The main contribution is that the Ebbinghaus Forgetting Curve [3] was introduced into the memory management of the vehicle dialogue system for the first time, and the anthropomorification attenuation and enhancement of machine memory were achieved through mathematical modeling. A complete dynamic graph update and pruning algorithm for EchoKG was proposed. The graph structure was dynamically adjusted through memory Strength and Rehearsal, significantly reducing the context load while ensuring personalization.

2. Related Work

Early long-term memory methods mainly relied on rule-based Slot Filling, storing and retrieving key information through predefined structured fields. However, this method has obvious limitations in terms of expressive power and generalization. With the rise of the Transformer architecture, the memory mechanism based on vector retrieval Augmented Generation (RAG) has gradually become mainstream. By storing historical dialogue summaries in vector databases and retrieving them based on semantic similarity, more flexible long-term dependency modeling has been achieved [4].

However, methods such as Memory Bank will lead to a decline in index efficiency over long-term operation due to the continuous accumulation of data volume, affecting the system response speed and quality. Works such as LongMem and LangMem have attempted to alleviate the problem of context redundancy through hierarchical storage and priority strategies [5], but they are still insufficient when dealing with changes in user preferences over time or

even instruction conflicts (such as users modifying previously given preferences).

Meanwhile, knowledge graphs have long been used to enhance the knowledge understanding of dialogue systems due to their structured expression and explicit reasoning capabilities. For example, K-BERT significantly improved the accuracy of domain knowledge question answering by injecting knowledge graph triples into the input layer [6]. However, the existing work generally focuses on general encyclopedic Knowledge (World Knowledge), and there is still a lack of systematic research on how to construct and maintain user profile graphs that can be continuously updated over time and reflect users' dynamic preferences, especially in highly personalized continuous interaction scenarios such as vehicles, where there is even a blank.

Furthermore, the exponential decay law of memory over time revealed by the Ebbinghaus forgetting curve has been used in recommendation systems to simulate user interest drift and has also been widely applied in the Spaced Repetition algorithm in educational software [7]. However, in the field of dialogue management of large models, there are no mature methods for applying it to dynamic memory pruning or priority reorganization yet. In conclusion, there is still much room for exploration in how to effectively integrate long-term memory, knowledge graphs, and human memory patterns to construct sustainable and evolving user-level dialogue memory [8,9].

3. EchoKG frame

The overall architecture of EchoKG is shown in Figure 1 (a sketch, only describing the logic), and the system as a whole is composed of three closely collaborating modules. Firstly, the memory encoder and writer is responsible for parsing the natural language input into a structured "entity-relations-attribute" triplet and initializing the memory strength for the newly written preference information, providing a basis for subsequent dynamic evolution. Secondly, the Dynamic KG Core is implemented based on Neo4j. It maintains preference nodes with attributes such as timestamps, access frequencies, and creation times, and performs reinforcement and forgetting operations on the graph based on users' interaction behaviors, enabling it to reflect the long-term trends and immediate changes of users' preferences. Finally, the memory retrieval and enhancement generator retrives several most relevant subgraphs from the graph in the dialogue based on the current query, linearizes them and injects them into the language model to construct context inputs with more personalized user characteristics.

In terms of user preference modeling, we have constructed a dynamic preference knowledge graph $G = (E, R, P)$, which includes a set of preference entities, a set of semantic relations,

and a set of dynamic attributes. For any preference node, we maintain its key attributes such as memory strength s, last access time t_{last}, recurrence times n, and creation time. Take temperature preference as an example. A typical preference record can be expressed as:

$$< User_{001}, PREFERS_TEMP, 24C, \{S, n, t_{last}, t_{create}\} >$$

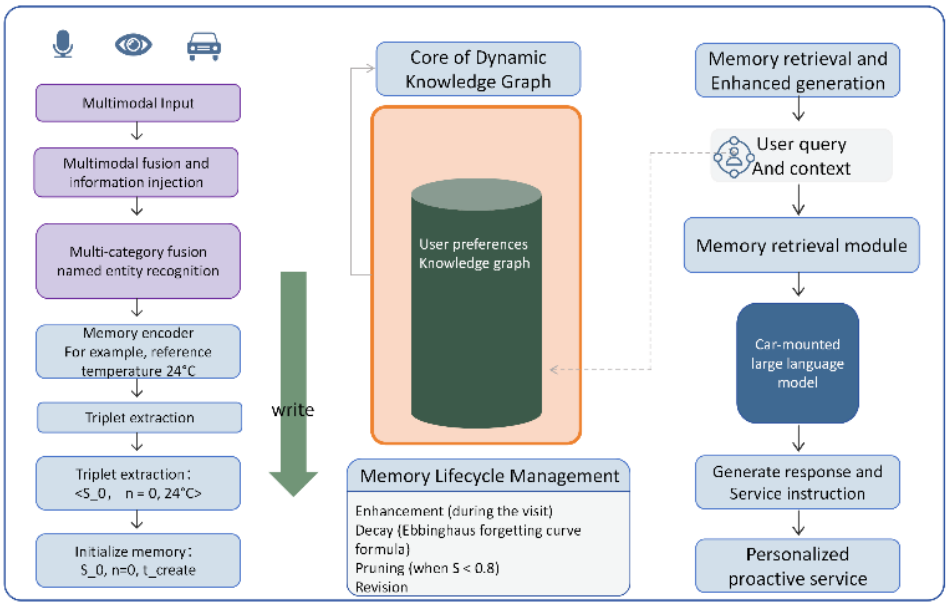

Figure 1:EchoKG framework Architecture diagram

The dynamic attributes among them are used to continuously describe the evolution state of preferences during the system's operation. When a user frequently mentions a certain preference, its memory strength will be enhanced, while when the preference remains inactive for a long time, it will naturally decline over time.

To simulate the forgetting mechanism of human memory, we combine the core idea of the Ebbinghaus forgetting curve and conduct a discrete modeling of it to adapt to the intermittent interaction mode in vehicle-mounted scenarios. In EchoKG, the temporal evolution of memory intensity depends on two key factors: one is the user's "review" behavior (i.e., the recurrence of preferences), and the other is the time interval since the last activation. Based on this, we update the memory intensity in the following form:

$$S(t) = f(n) \cdot g(\Delta t)$$

Here, $f(n)$ represents the enhancement effect that occurs with the increase in the number of reproductions, showing a marginal diminishing characteristic; And $g(\Delta t)$ depicts the exponential decay process of memory over time. To provide a more explicit modeling form, we parameterized it in the experiment, making the memory attenuation more in line with the usage frequency and interest change patterns of real users:

$$S(t) = S_0(1+n)^{\alpha} e^{-\beta \Delta t}$$

Here, S_0 represents the initial intensity, α controls the strengthening rate, and β describes the attenuation rate. In this way, the system can automatically achieve the effect of "retaining important preferences for a long time and gradually fading outdated preferences" in long-term interaction.

Overall, EchoKG effectively combines structured preference modeling, dynamic graph updates mechanisms, and human memory patterns, enabling the system to maintain personalized consistency while flexibly adapting to the natural changes in user interests. As a result, it demonstrates higher stability and intelligence in long-term interaction scenarios such as in-vehicle conversations.

The retrieval module uses Cypher query statements to obtain nodes with $S > 1.0$ and the Top -10 semantic similarity. The retrieved subgraphs are linearized into natural language prompt words. For example: Prompt: "User historical preference memory: [Air Conditioning temperature: 24 degrees (Strong preference)], [Frequently Heard singer: Eason Chan (Medium preference)]. Please reply to the user based on this".

4. Experiments

4.1. Dataset Construction

To address the long-standing problem of scarce public data in the field of in-vehicle dialogue, we have built and open-sourced the EchoCar-Public dataset. Based on the systematic cleaning, integration and reconstruction of the existing multi-round dialogue resources, this dataset generates supplementary long-term preference scenarios through a large model, and finally forms a Chinese-English mixed dataset containing 15,800 rounds of dialogues. Among them, the English part is mainly derived from typical task-oriented corpora covering transportation, navigation and ancillary services such as MultiWOZ 2.4, SGD and KVRET [11-13]; The Chinese part integrates Chinese MultiWOZ and CarChat-1K, and utilizes approximately 5% of the large model to enhance the samples and expand the diversity of cross-round preference expressions and temporal dependencies. To evaluate the adaptability and forgetting mechanism of the model in long-term interaction, we deliberately injected preference conflict and correction events spanning different time spans (such as Day 1, Day 7, Day 30) into the dialogue, enabling the dataset to more comprehensively cover preference drift behavior in real scenarios.

4.2. Experimental Setup

The experiment was carried out based on Qwen2-7B-Chat (4-bit quantization), and vector retrieval memory banks, static knowledge graph structures, long-term memory compression methods, and commercial closed-source memory mechanisms were selected as control schemes to comprehensively investigate the differences in efficiency, accuracy, and stability of different memory systems in vehicle scenarios. To achieve more identifiable comparisons, we comprehensively measure system performance by using indicators such as intent recognition F1, context length, personalized consistency, and response delay [14]. The degree of intent recognition reflects the semantic understanding ability of the model. The length of the context reflects the compression ability of different memory strategies on the input scale of LLMS. Personalized consistency is used to verify whether the response aligns with the user's historical preferences. Response delay measures the availability of a system in real-time interaction.

4.3. Main Results

The experimental results show that EchoKG demonstrates significant advantages in both efficiency and long-term stability. In terms of context management, as the graph can compress the original dialogue into discrete and structured preference nodes, the number of input tokens generated by EchoKG is only about half of that of traditional vector retrieval schemes, thereby significantly reducing the model inference cost and keeping the response delay at an acceptable low level for in-vehicle interaction. In terms of semantic understanding, the dynamic forgetting mechanism effectively eliminates outdated preferences, reduces noise interference, and makes the intent recognition performance superior to that of static graphs. It is also worth noting that in terms of the personalized consistency index evaluated manually, the performance of EchoKG is close to that of commercial closed-source memory systems, indicating that the introduction of a time decay mechanism helps the model form a preference retention behavior similar to human "familiarity" in long-term interactions.

To further verify the long-term stability of the system, we constructed a 30-day simulated interaction scenario. The results show that traditional static graphs will continuously accumulate one-off preferences in the early stage, leading to structural redundancy. Over time, EchoKG will gradually weaken the memory intensity of low-frequency preferences and automatically perform pruning operations when the intensity drops below the threshold, keeping the scale of the spectrum always within a controllable range and being able to dynamically reflect the user's true long-term habits. This phenomenon verifies the rationality of modeling based on the Ebbinghaus forgetting curve and also indicates that introducing

psychological memory laws into the graph memory system has dual advantages in theory and practice.

Table 1. The experimental results.

Method	Intention F1	Token	Personalized consistency (1-5)	MOS	Delay (ms)
Vanilla Qwen2	0.796	1980	2.58	3.34	670
MemoryBank	0.837	2980	3.71	3.91	1280
Static KG	0.854	1820	4.05	4.12	710
EchoKG (Ours)	0.905	1340	4.73	4.79	780
GPT-4o Memory	0.918	-	4.81	4.86	2200+

5. Discussion and Limitations

While introducing a forgetting mechanism to enhance system efficiency, the high safety requirements of in-vehicle scenarios also impose additional constraints. For important information related to driving safety or emergency response, such as users' preferences for vehicle handling characteristics (such as brake sensitivity), emergency contacts, etc., their semantic attributes have a high degree of safety sensitivity and thus should not be weakened over time. Based on this, we designed and implemented the "Immortal Whitelist" mechanism in EchoKG, forcibly setting the attenuation coefficient beta to 0 for all attributes marked as Safety-Critical. Theoretically, it is necessary to ensure that such information has permanent memory weights in the graph, thereby achieving the non-forgeability of security semantics.

On the other hand, the parameters alpha and beta in the forgetting curve have a decisive influence on the memory evolution process, and the preference patterns of different user groups may vary significantly in the time dimension. For instance, the preference switching frequency of young users is usually higher, which implies that a larger attenuation coefficient beta may be required in dynamic modeling. In contrast, elderly users with more stable preferences correspond to a slower rate of memory decline. The above phenomena indicate that fixed parameters are difficult to cover the heterogeneity of the real user group. Therefore, future work will extend to the parameter adaptive method based on Meta-Learning [15], enabling the forgetting model to continuously adjust according to the long-term behavioral characteristics of users, thereby achieving more refined personalized memory management.

In addition, the current computing of EchoKG is mainly deployed at the edge nodes of the vehicle to ensure that the inference delay meets the real-time requirements of in-vehicle

interaction. However, the computing resources at the vehicle end are limited, while large-scale graph construction, attribute clustering, and cross-user knowledge mining are more suitable to be carried out in the cloud where resources are abundant. Therefore, we plan to further explore the "vehicle-cloud Federation" collaborative architecture: completing high-complexity graph enhancement and statistical modeling on the cloud side, and performing lightweight inference and local storage of privacy-sensitive information on the vehicle side, thereby achieving cross-terminal knowledge fusion and dynamic synchronization while ensuring user privacy and system efficiency.

6. Conclusions

The EchoKG framework proposed in this paper innovatively utilizes the Ebbinghaus forgetting curve to solve the problem of long-term memory management in in-vehicle dialogue systems. Through mathematical modeling with dynamic weights, EchoKG significantly reduces computing resource consumption and response delay while maintaining high-precision personalized services. Experimental data show that this method has extremely high practical value in real vehicle scenarios.

References

[1] Murali, P. K., Kaboli, M., & Dahiya, R. (2022). Intelligent in-vehicle interaction technologies. *Advanced Intelligent Systems*, *4*(2), 2100122.

[2] Zhong, W., Guo, L., Gao, Q., Ye, H., & Wang, Y. (2024, March). Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 17, pp. 19724-19731).

[3] Memory, O. K. C. Memory: A Contribution to Experimental Psychology.

[4] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, *33*, 9459-9474.

[5] Packer, C., Fang, V., Patil, S., Lin, K., Wooders, S., & Gonzalez, J. (2023). MemGPT: Towards LLMs as Operating Systems.

[6] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020, April). K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 03, pp. 2901-2908).

[7] Settles, B., & Meeder, B. (2016, August). A trainable spaced repetition model for language learning. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 1848-1858).

[8] Park, J. S., O'Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023, October). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology* (pp. 1-22).

[9] Trivedi, R., Dai, H., Wang, Y., & Song, L. (2017, July). Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning* (pp. 3462-3471). PMLR.

[10] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., ... & Zhu, T. (2023). Qwen technical

report. *arXiv preprint arXiv:2309.16609*.

[11] Budzianowski, P., Wen, T. H., Tseng, B. H., Casanueva, I., Ultes, S., Ramadan, O., & Gašić, M. (2018). Multiwoz--a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

[12] Rastogi, A., Zang, X., Sunkara, S., Gupta, R., & Khaitan, P. (2020, April). Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 05, pp. 8689-8696).

[13] Eric, M., Krishnan, L., Charette, F., & Manning, C. D. (2017, August). Key-value retrieval networks for task-oriented dialogue. In *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue* (pp. 37-49).

[14] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

[15] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., ... & Larson, J. (2024). From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Research on Bio-inspired Self-balancing Control Based on LIF Network

Zhixin Yan, Jin Li*, Junbang Jiang, Shanmengdai Luo, Lifang Huang

School of Science, Hubei University of Technology, Wuhan, Hubei, P.R. China

Received: January 18, 2026

Revised: January 19, 2026

Accepted: January 23, 2026

Published online: January 23, 2026

To appear in: *International Journal of Advanced AI Applications, Vol. 2, No. 2 (February 2026)*

* Corresponding Author: Jin Li (10279323@qq.com)

Abstract. Human balance is a skill gradually established through a sensory-action-feedback loop, relying on repetitive training, trial-and-error mechanisms, and the dynamic plasticity of synaptic connections. In this process, sensory signals are continuously transmitted to the central nervous system, where stable motor paths are formed through learning, enabling action reuse without complex calculations. Inspired by this mechanism, this paper proposes a balance learning method based on brain-like spiking neural networks and dopamine-modulated synaptic plasticity for self-learning control of the classic inverted pendulum system. The method connects the one-hot encoded sensory neuron group with motor neurons and utilizes a reward-driven synaptic weight update mechanism to gradually master the stable control of the inverted pendulum without the need for prior models or training data. Unlike traditional control algorithms such as PID or LQR, this approach features biological realism, strong adaptability, and self-organizing behavior, providing a new perspective on bio-inspired learning strategies for artificial intelligence in continuous control tasks.

Keywords: *Spiking Neural Network; Dopamine-modulated Synaptic Plasticity; Autonomous learning; Reward*

1. Introduction

In traditional control engineering, control loops typically consist of several key modules: the internal and external state perception modules, the control decision module, and the system dynamic model module [1, 3]. The working principle of a controller is to predict the future expected state based on the system's current state and the acquired environmental information, and then generate control actions accordingly, ultimately driving the system to achieve the

desired behavior. In terms of control methods, model-based control relies on accurate modeling and simulation of the physical process to predict system behavior, while model-free control does not require an explicit dynamic model [4, 14], instead optimizing the control strategy through continuous interaction with the environment.

In recent years, machine learning techniques, particularly deep reinforcement learning (Deep Reinforcement Learning, DRL), have been widely applied in control tasks such as industrial process control, autonomous driving decision-making, and robotic operations, due to their powerful ability to learn complex strategies in high-dimensional state spaces [7, 12]. However, although traditional artificial neural networks (ANNs) mimic the connections of biological neurons in structure, their computational units are essentially continuous numerical mappings. This fundamentally differs from the time-dependent computational mechanisms that biological neural systems rely on, which depend on spike transmission [13, 18-23]. To bridge this gap, Spiking Neural Networks (SNNs), as the "third generation of neural networks," have been proposed. SNNs use spike trains in the time domain to transmit information, more accurately simulating the way signals are transmitted between biological neurons [9, 10]. The advantage of SNNs lies not only in their ability to encode information in time, but also in their event-driven sparse activation mechanism, which significantly improves energy efficiency, making them more suitable for embedded control scenarios with limited resources.

To enable SNNs to learn effective control strategies, researchers have developed various reward-modulated synaptic plasticity mechanisms. For example: R-STDP (Reward-modulated STDP): Combines the spike-timing differences (STDP) of pre- and post-synaptic spikes with external reward signals to achieve fine-tuning optimization of the strategy.RM-STDP: Builds upon R-STDP by introducing a weight-dependent multiplicative modulation factor to enhance the stability of the training process and the generalization ability of the strategy [9, 24-27].TD-STDP: Introduces the temporal difference error from reinforcement learning into the synaptic learning process and uses an eligibility trace mechanism to address the reward delay issue.

Although mechanisms such as R-STDP, DA-STDP, and TD-STDP have initially established a connection between synaptic plasticity and environmental rewards, they still have limitations in terms of biological realism, effective handling of delayed rewards, and adapting to dynamic task feedback. R-STDP mainly controls and amplifies the synaptic update based on instantaneous reward signals, making it difficult to effectively cope with situations where reward signals are significantly delayed [16, 17]. The DA-STDP model only establishes a weight update mechanism between pre- and post-synaptic spikes and fails to capture delayed

rewards that appear several seconds after the behavior [28-32].

In contrast, DE-STDP (Dopamine-Eligibility STDP) shows greater potential in terms of biological plausibility and mechanism consistency [8, 33]. This mechanism uses dopamine (DA) concentration as a dynamic modulation factor and introduces the "eligibility trace" variable, coupling the local plasticity of STDP with the global reward signal reflected by dopamine concentration, giving synaptic weight changes "causal controllability" over time. This not only naturally simulates the core function of dopamine in reward-driven learning in biological neural systems, but also eliminates the need for external TD error calculation modules. The key feature of DE-STDP lies in its temporally separated weight update mechanism: STDP determines the possible direction of weight change based on spike timing differences (eligibility trace). The reward gating is then executed, with dopamine signals deciding whether these preset changes are actually implemented. This "trace-reward" pairing mechanism aligns with the time-scale differences between plasticity events and reward signals in biological systems [11, 15]. This two-stage regulation strategy makes DE-STDP advantageous in tasks involving sparse reinforcement signals, significant reward delays, or the need for local plasticity adjustments.

Unlike current mainstream control methods based on reinforcement learning or deep neural networks, this study emphasizes exploring the synaptic learning rules and biological information processing mechanisms achievable by the nervous system itself, and focuses on the possibility of efficient, unsupervised balance learning in low-dimensional state spaces. The research not only validates the practical feasibility of DE-STDP in dynamic control tasks but also provides theoretical foundations and potential technical pathways for promoting brain-like computational paradigms in practical control systems.

2. Methodology

2.1 Network Structure

To achieve reinforcement learning control for the inverted pendulum system, this study constructs a two-layer spiking neural network consisting of an input layer and an output layer. The network structure is simple, with clear connections, providing good biological interpretability and hardware deployment potential.

The input layer consists of 24 Leaky Integrate-and-Fire neurons, which receive discretized encoded information of the environment's state. Specifically, the system's four-dimensional state variables (cart position, cart velocity, pole angle, and angular velocity) are discretized into several intervals and mapped to the 24 neurons using one-hot encoding. This ensures the

unambiguous transmission of state information and the capability for spike-based expression. The output layer contains 2 neurons, each representing one of the two discrete control actions (applying force to the left or applying force to the right). The network uses a fully connected structure, meaning each neuron in the input layer is synaptically connected to all neurons in the output layer.

To reduce computational complexity and enhance the biological plausibility of neuron behavior, this study adopts the classic Leaky Integrate-and-Fire model for neuron modeling [37-39]. In this model, each neuron contains only one state variable—its membrane potential $V(t)$, and its dynamic behavior follows the differential equation:

$$\frac{dV}{dt} = -\frac{V(t) - V\_rest}{\tau\_m} + \frac{I\_syn(t) + I\_ext(t)}{C\_m}$$

In this model, V_rest represents the resting potential, τ_m is the membrane time constant, and C_m is the membrane capacitance. $I\_ext(t)$ represents the externally injected current, primarily coming from the state perception input. $I\_syn(t)$ is the total synaptic current, triggered by synaptic inputs from within the network. When the membrane potential $V(t)$ exceeds the threshold voltage V_th, the neuron is considered to fire a spike and undergoes a potential reset followed by a refractory period [4].

This network architecture fully integrates the fundamental characteristics of biological neural systems, while maintaining high engineering feasibility, providing a solid foundation for subsequent control learning based on reward-modulated spiking plasticity rules.

2.2 State Discretization and One-Hot Encoding

The spikes generated by the input neurons are used to encode the observation states of the inverted pendulum system. Each observation variable of the system (including the cart position x、velocity v、pole angle θ and angular velocity ω) is mapped to an integer index according to the following rule[32]：

$$id\_obs = \begin{cases} 0, & obs \le obs_{min} \\ floor(\frac{x - x_{min}}{\Delta x}), & obs_{min} < obs < obs_{max} \\ N_{states,obs} - 1, & obs \ge obs_{max} \end{cases}$$

In this context， Δx is the width of each interval, and obs_{min} and obs_{max} are the discretization limits for the variable. The total number of discrete states for each variable is given by： $N_{states,obs} = ceil(\frac{x - x_{min}}{\Delta x})$ ， The combination of the four observation variables forms a complete state $(id_x, id_v, id_\theta, id_\omega)$， The total number of states in the system is:

$$N_{states,total} = N_{states,x} * N_{states,v} * N_{states,\theta} * N_{states,\omega}$$

To achieve a unique representation for each state, each group of states is encoded by a set of n_{input} input neurons. Therefore, the total number of neurons in the input layer of the SNN is:

$$N_{input\ neurons} = N_{states,total} * n_{input}$$

When a specific state is input, only the n_{input} neurons corresponding to that state will spike, while all other neurons remain silent. This method is a classic example of one-hot encoding [30,34], which is commonly used in machine learning to represent categorical variables. For the discretization of the angle θ：

the central balanced region [-π/12, π/12]（equivalent to [-15°, 15°]）is divided into 10 subintervals；

The other unbalanced regions (such as [-π/2, -π/12] and [π/12, π/2]) are divided into coarser subintervals.

This type of "sparse-dense-sparse" partitioning helps to enhance the system's resolution in the critical balanced region, thereby improving control performance.

2.3 Reward Function Design

Intuitively, the reward function should reflect the core objective of the control task, which is to maintain the pole in the upright position. Since the control outcome depends on the action selected and executed in the current state of the system, when an action guides the system toward a direction more favorable for achieving this goal, it should be assigned a positive reward. To enhance the Spiking Neural Network (SNN) controller's responsiveness to system dynamics, various reward functions are designed based on the evolution of the state. As the reward function progresses from R_1 to R_2， the perceptual variables introduced become more complex, and the feedback mechanism transitions from a single physical quantity to a composite trend judgment. This allows the system to become more sensitive to "balance tendency" during the training process [35,40]. The second reward function R_1 is based on the trend of angular velocity changes between two time steps.

$$R_1(\omega_{old},\omega_{new}) = \begin{cases} 1, & \omega_{old} * \omega_{new} < 0 \\ 1, & |\omega_{new}| > |\omega_{old}| \\ -1, & otherwise \end{cases}$$

In this context, the first term checks whether the direction of the angular velocity has reversed, which indicates that the system is attempting to correct the existing rotational trend. The second term encourages a reduction in angular velocity, reflecting the control action's effect in

suppressing the rotation amplitude. If neither of these conditions is met, the action is considered ineffective, and a punitive reward of -1 is applied to the system.

R_2 builds upon R_1 by further considering the trend in the direction of the angle to improve the system's overall ability to judge the return to equilibrium. It is defined as follows:

$$R_2(\omega_{old},\omega_{new},\theta_{old},\theta_{new})=\begin{cases} R_1(\omega_{old},\omega_{new}), & \theta_{new}*\omega_{old}>0 \\ 1, & \theta_{new}*\omega_{old}\leq 0 \text{ and } \theta_{new}*\omega_{new}<0 \\ -1, & otherwise \end{cases}$$

The logic of this function emphasizes that when both the angular velocity and the angle direction point toward the "return to vertical" trend, a positive reward should be given; otherwise, a penalty is applied. Particularly in some cases, if the angle θ_{old} and the angular velocity ω_{old} have opposite signs, it indicates that the current angular velocity is actually decreasing the tilt angle, meaning the action itself has a positive effect. In such a case, simply using the "direction reversal or deceleration" criterion in R_1 is insufficient to accurately evaluate the system's evolution. Therefore, R_2 further introduces a check on the sign combination of θ_{new} and ω_{new}:if the signs of θ_{new} and ω_{new} are opposite, it indicates that the new state is still maintaining the ideal trend of "angular velocity correcting the angle," and a positive reward is given; otherwise, the action is considered detrimental to system balance, and a punitive reward of -1 is applied. Compared to R_1, R_2 can more accurately recognize the actual contribution of the agent's action to the "system's return to balance" and provides more directional feedback signals during the SNN learning process.

2.4 DE-STDP

Since the dynamics of intracellular processes triggered by STDP and dopamine (DA) are not yet fully understood, this paper proposes a simplified phenomenological model to characterize the basic mechanism by which DA regulates STDP plasticity. Referring to the method by *i* et al. (2004) [46], the paper uses two phenomenological variables to describe the state of each synapse: the synaptic weight (s) and the enzyme activity variable (c) closely related to synaptic plasticity, such as the autophosphorylation of CaMK-II (Lisman, 1989), oxidation reactions of PKC or PKA, or other slower biochemical processes. These processes together form the so-called "synaptic tag" [38-41].

The basic dynamics of the model are described as follows:

$$\dot{c}=-\frac{c}{\tau_c}+STDP(\tau)\delta(t-t_{pre/post})$$

Here, *(\delta(t))* is the Dirac delta function, which is triggered when the pre- or post-neuron fires at the times *(t_{\text{pre}})* or *(t_{\text{post}})*, causing the variable (*c*) to be updated

according to the STDP curve (Figure 1b). To clarify the mathematical nature of the STDP mechanism, the following model function is used to describe the synaptic timing-dependent plasticity changes [2, 47]:

$$W(\Delta t)\begin{cases} A^{+}e^{(-\frac{\Delta t}{\tau^{+}})}, & if\ \Delta t>0 \\ -A^{-}e^{(\frac{\Delta t}{\tau^{-}})}, & if\ \Delta t<0 \end{cases}$$

$\Delta t=t_i-t_j$ represents the time difference between the postsynaptic and presynaptic neuron spikes, with A^+ and A^- representing the maximum adjustment amplitudes for long-term potentiation (LTP) and long-term depression (LTD), respectively, and τ^+ 、 τ^- being the corresponding time window constants. This function characterizes the update magnitude of the synapse at different time differences, reflecting the fundamental principles of STDP.

The accumulated "plasticity potential" of the variable ccc only influences the synaptic weight sss when the DA concentration d > 0, enabling synaptic strengthening or weakening. Therefore, *c(t)* is considered as the "plasticity trace" or "eligibility trace" of the synapse, a concept introduced by Houk, Adams, and Barto (1995) [43-46]. Additionally, the dynamics of DA are described by the following equation:

$$\dot{d}=-\frac{d}{\tau_d}+DA(t)$$

Here, τ_d is the dopamine (DA) uptake time constant, and DA(t) represents the DA input generated by dopaminergic neuron firing in brain structures such as the ventral tegmental area (VTA) and the substantia nigra compacta. In this study, $\tau_d = 0.01$ s， s is set to reflect the rapid clearance of DA in physiological processes. To better simulate the phasic and tonic patterns of DA, and in line with the dopamine encoding logic shown in Figure 1, when the system receives a reward (reward = 1), *DA(t)* is set to 0.05 μM, corresponding to the phasic activation triggered by reward in Figure 1(a) or the activation after conditioned stimulus predicts a reward in Figure 1(b). In the absence of a reward or with a negative reward (reward = -1), DA(t) is maintained at a baseline level of 0.001 μM, corresponding to tonic inhibition during the reward absence shown in Figure 1(c). At the same time, the background DA concentration is incorporated into the STDP weight update mechanism, represented by the following formula:：

$$\dot{s}=c(d-d\_baseline)$$

Here, *d_baseline* = 0.005 μM represents the background DA level of the system. This mechanism makes the synaptic potentiation process more sensitive to increases in DA concentration, while it becomes less likely to produce reinforcement effects when the DA level

is below the baseline, helping to suppress the phenomenon of false reinforcement.

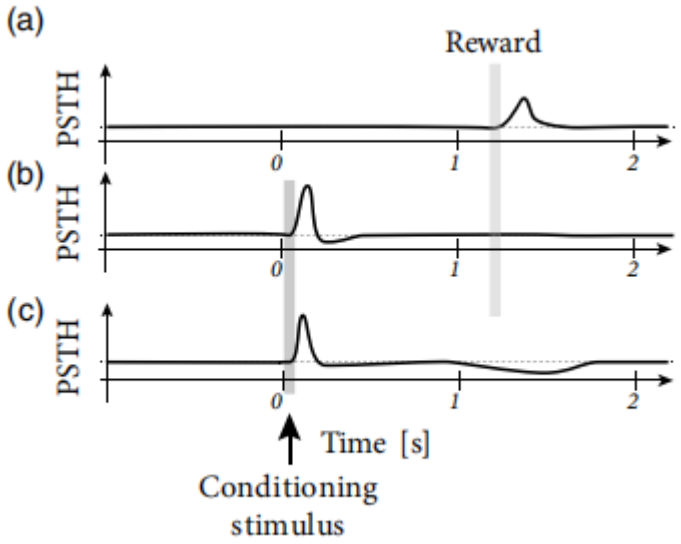

Figure 1. dopamine reward rule

In the inverted pendulum control system, the learning and reward mechanism is similar to the dopamine response logic shown in Figure 1. When the system successfully maintains balance, it corresponds to the reward activation in Figure 1(a), where dopamine activity in the neurons increases, reinforcing the successful balancing action. As the system learns, if the inverted pendulum has already learned the relationship between specific control signals and successful balance, these signals become conditioned stimuli, similar to the situation in Figure 1(b), where neurons respond to the conditioned stimulus in advance, without waiting for the reward to arrive. Eventually, when the system can predict the reward through the conditioned stimulus, the neuron's response becomes more stable, as shown in the trough in Figure 1(c), indicating that the system has learned how to efficiently and automatically maintain balance, without relying on every reward feedback. This learning process makes the inverted pendulum system more independent, enabling it to maintain balance more stably.

In summary, the model reasonably integrates the millisecond-scale synapse-specific STDP with the second-scale behavioral feedback in terms of timescale differences, as reflected in the dopamine encoding of reward timing in Figure 1. Although there is currently no direct experimental evidence to prove or disprove this model, it provides a clear, testable theoretical framework for exploring the regulatory mechanism of DA in STDP.

3. Results

3.1 Experimental Environment

The Cart-Pole system is one of the most classic control problems in reinforcement learning and is widely used to evaluate the performance of various control algorithms. In recent years, many studies based on Spiking Neural Networks (SNNs) have also used this system as a platform for algorithm testing [35,42]. This task can be described as follows: a cart and a rod connected by a hinge form the system, with the rod being able to rotate only in the plane perpendicular to the ground. The cart (Fig. 2) moves along a frictionless horizontal track, and the control agent must choose an action in each frame: apply a force to the left or to the right. The chosen action will affect the dynamics of the entire system, with the control objective being to keep the rod upright for as long as possible without becoming unstable.

In the MuJoCo simulation environment, decisions are made every 16 milliseconds. The observed system state includes:

The position of the cart: x, in meters;

The velocity of the cart: $v = \frac{dv}{dt}$, in meters per second;

The angle of the rod: θ, in radians (usually referenced to the vertical direction);

The angular velocity of the rod: $\omega = \frac{d\theta}{dt}$, in radians per second.

The simulation will terminate when any of the following conditions are triggered:

Rod tilt: The absolute value of the rod's angle exceeds 15°.

Cart out of bounds: The position of the cart exceeds the track boundaries of -2.0 meters to 2.0 meters.

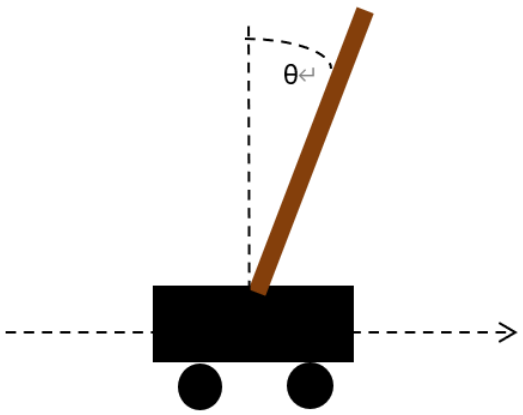

Figure 2. Cart and pole

3.2 Experimental Plan

In this experiment, the initial network weights are set to small random values, and the Spiking Neural Network (SNN) learns online through continuous interaction with the environment. The system's learning objective is to continuously keep the rod within a specified angle threshold range, i.e., in the "balanced state," for each episode until the cart exceeds the track boundary, which is considered a successful episode. The training process consists of 200 episodes. To evaluate the model's stability and generalization ability within a local time window, this paper introduces a sliding window success rate metric. Specifically, it is defined as the proportion of episodes within a sliding window of fixed length (20 episodes) where the number of balanced steps exceeds 7000 steps. This metric is considered the probability of "success" within the window. It dynamically reflects the phase effectiveness of the strategy and the stability improvement during the convergence process. To comprehensively evaluate the performance of different STDP mechanisms, all employing the reward function defined in R_2 ,the experiment compares the training performance of three plasticity rules: R-STDP (basic version), DA-STDP (with dopamine signal), and DE-STDP (with error and dopamine signal).

3.3 Experimental Results and Analysis

3.3.1 Evolution of Balance Steps During Training

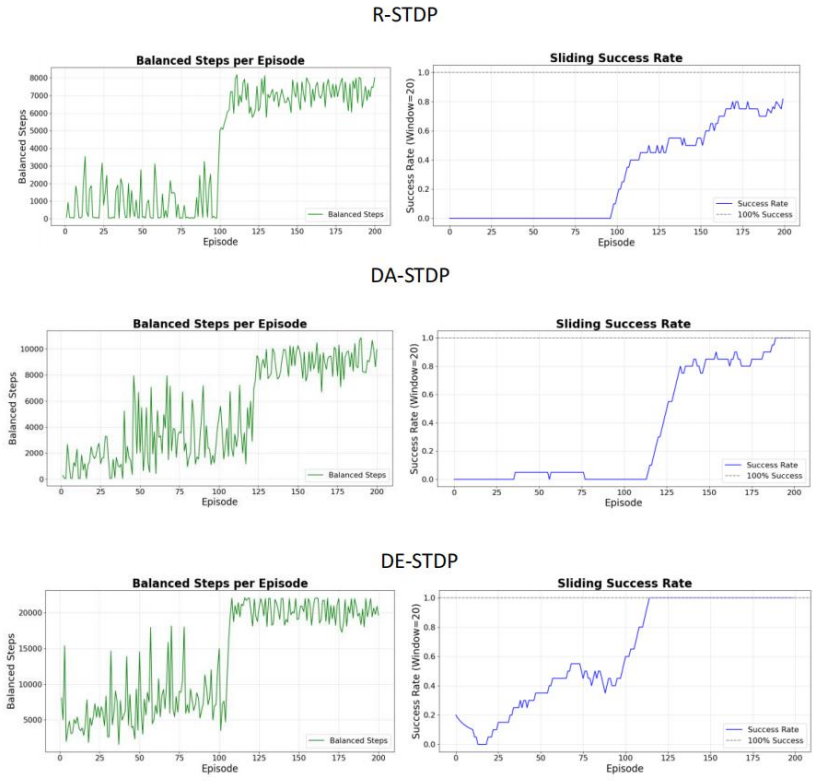

Figure 3. the comparison of performance for the three different STDP mechanisms

Fig. 3 shows the evolution of the number of balance steps per episode during the training process under three different STDP learning rules. R-STDP exhibits a significant training delay, with a notable improvement occurring only after around the 100th episode. In contrast, DA-STDP and DE-STDP quickly converge around the 110th episode, with DE-STDP demonstrating a strong learning capability in the early stages and maintaining the highest stability after convergence.

As shown in the figure, under DE-STDP modulation, the number of balance steps in the SNN during the CartPole task evolves over the course of training. In the initial phase (approximately the first 100 episodes), the SNN struggles to maintain the rod's stability, demonstrating a clear exploration phase. However, as training progresses, the synaptic connections are gradually optimized under DA modulation, and the system's balancing ability improves significantly. DE-STDP outperforms both R-STDP and DA-STDP in terms of convergence speed and stability, while DA-STDP shows a higher success rate and better sustained balance ability compared to R-STDP in the later stages.

3.3.2 Evolution of Maximum Angle During Training

This experiment uses the "maximum angle per episode" as a core observation metric to compare the training performance of R-STDP, DA-STDP, and DE-STDP in reinforcement learning tasks. By analyzing the fluctuations of the maximum angle over 200 episodes, the convergence and stability of different mechanisms are evaluated. From the experimental curves, the performance differences among the three STDP mechanisms are significant: R-STDP remains within a large oscillation range of -15° to 15° throughout the 200 episodes, with the system continuously cycling between "exploration and loss of control." This occurs because it relies solely on the temporal correlation between pre- and post-synaptic neurons, without considering "reward delay" or "error feedback," leading to an inability to establish a stable "action-reward" relationship. Its variance is 112.39, indicating large fluctuations.

DA-STDP, through dopamine encoding of the "reward prediction error," shows phase-wise convergence. The fluctuations in the first 50 episodes are similar to R-STDP, but after the 75th episode, the oscillation amplitude gradually decreases. After the 125th episode, it stabilizes between -5° and 10°. Although there is some convergence, due to the unresolved "temporal mismatch between actions and delayed rewards," there is still some fluctuation in the later stages. Its variance is 116.38, with reduced volatility compared to R-STDP.

DE-STDP performs the best. There is some fluctuation in the first 50 episodes, but after the 75th episode, the oscillation amplitude rapidly narrows. After the 125th episode, it stabilizes

between -5° and 5°, and approaches 0°, achieving stable angle control. Its variance is 55.62, indicating a more stable learning process. Overall, R-STDP performs the worst due to the lack of adaptation to reward delay, DA-STDP shows improvement but with limited convergence, and DE-STDP excels in both convergence speed and stability, providing a more efficient STDP-based reinforcement learning framework.

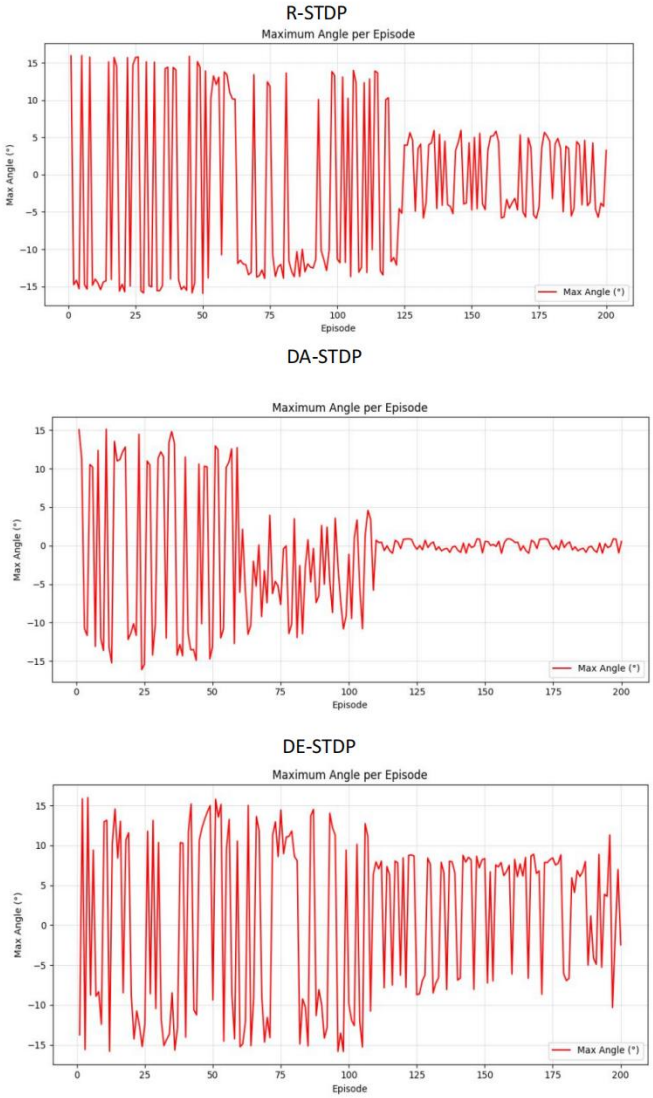

Figure 4. shows the comparison of performance for the three different STDP mechanisms, illustrating the fluctuations of the maximum angle over 200 episodes.

3.4 Summary

This paper presents and implements a biologically-inspired phenomenological modeling approach focused on dopamine-modulated, time-dependent synaptic plasticity mechanisms, aiming to explain how delayed rewards at the behavioral level can lead to adjustments in synaptic strengths at the neural synapse level. The model draws from the ideas proposed by

Izhikevich et al., with the core concept being the introduction of two synaptic variables: synaptic weight (s) and the eligibility trace variable (c). The model is biologically grounded, combining the weight potential change (STDP rule) triggered by spikes with the delay mechanism of reward signals. This method is particularly suited to address a common issue in reinforcement learning — the delay of rewards relative to the timing of neural firing behaviors.

Additionally, the DA signal in the model is expressed in both baseline and phasic forms, with the sensitivity of weight adjustments under different DA concentrations enhancing the system's ability to differentiate environmental feedback and avoid erroneous reinforcement. This strategy effectively resolves the insensitivity to delayed rewards found in traditional STDP models, offering enhanced learning stability and biological plausibility. In conclusion, this approach provides a reasonable and experimentally testable modeling framework for synaptic learning mechanisms in neuromorphic reinforcement learning, especially suited for adaptive behavioral learning systems in delayed reinforcement scenarios.

Acknowledgements

Guiding project of Scientific Research Plan of Hubei Provincial Department of Education, Design of Application-Specific Integrated Circuit for Voice Command Recognition Based on Compute-in-Memory Architecture (No. B2020046); Cooperation Agreement on the Project of Design and Process Manufacturing Technology of 3D Stacked Code Flash Memory Chip (No. 2021802); PhD Research Foundation Project of Hubei University of Technology, Research on Sun Tracking Control and System State Monitoring of Butterfly Concentrating Photovoltaic Based on Deep Learning Image Analysis (No. 00185).

ORCID:

Jin Li - https://orcid.org/0000-0002-4615-2574

Junbang Jiang - https://orcid.org/0009-0008-8914-5658

References

[1] Baydin, A. G., Pearlmutter, B. A., Syme, D., Wood, F., & Torr, P. (2022). Gradients wit hout backpropagation. *arXiv preprint arXiv:2202.08587*.

[2] Burms, J., Caluwaerts, K., & Dambre, J. (2015). Reward-modulated Hebbian plasticity a s leverage for partially embodied control in compliant robotics. *Frontiers in neuroroboti cs*, *9*, 9.

[3] Barto, A. G. (2019). Reinforcement learning: Connections, surprises, and challenge. *AI Magazine*, *40*(1), 3-15.

[4] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zarem ba, W. (2016). Openai gym. arXiv preprint arXiv:1606.01540.

[5] Chen, W. (2022). Neural circuits provide insights into reward and aversion. Frontiers in Neural Circuits, 16, 1002485.
[6] Chevtchenko, S. F., Bethi, Y., Ludermir, T. B., & Afshar, S. (2024, June). A neuromorph ic architecture for reinforcement learning from real-valued observations. In 2024 Internat ional Joint Conference on Neural Networks (IJCNN) (pp. 1-10). IEEE.
[7] Barto, A. G., Sutton, R. S., & Anderson, C. W. (2012). Neuronlike adaptive elements tha t can solve difficult learning control problems. IEEE transactions on systems, man, and c ybernetics, (5), 834-846.
[8] Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. Current opinion in neurobiology, 10(6), 732-739.
[9] Gerstner, W., Kistler, W. M., Naud, R., & Paninski, L. (2014). Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press.
[10] Gewaltig, M. O., & Diesmann, M. (2007). Nest (neural simulation tool). Scholarpedia, 2 (4), 1430.
[11] Jitsev, J., Abraham, N., Morrison, A., & Tittgemeyer, M. (2012, September). Learning fr om delayed reward und punishment in a spiking neural network model of basal ganglia w ith opposing D1/D2 plasticity. In International Conference on Artificial Neural Network s (pp. 459-466). Berlin, Heidelberg: Springer Berlin Heidelberg.
[12] Mathews, M. A., Camp, A. J., & Murray, A. J. (2017). Reviewing the role of the efferent vestibular system in motor and vestibular circuits. Frontiers in Physiology, 8, 552.
[13] Markov, B., & Koprinkova-Hristova, P. (2024, September). Reinforcement Learning Co ntrol of Cart Pole System with Spike Timing Neural Network Actor-Critic Architecture. In International Conference on Artificial Intelligence: Methodology, Systems, and Appli cations (pp. 54-63). Cham: Springer Nature Switzerland.
[14] Shim, M. S., & Li, P. (2017, May). Biologically inspired reinforcement learning for mob ile robot collision avoidance. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 3098-3105). IEEE.
[15] Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP an d dopamine signaling. Cerebral cortex, 17(10), 2443-2452.
[16] Akl, M., Sandamirskaya, Y., Ergene, D., Walter, F., & Knoll, A. (2022, July). Fine-tunin g deep reinforcement learning policies with r-stdp for domain adaptation. In Proceedings of the International Conference on Neuromorphic Systems 2022 (pp. 1-8).
[17] Bing, Z., Meschede, C., Huang, K., Chen, G., Rohrbein, F., Akl, M., & Knoll, A. (2018, May). End to end learning of spiking neural network based on r-stdp for a lane keeping v ehicle. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 4 725-4732). IEEE.
[18] Han, Z., Chen, N., Xu, J., & Li, W. (2021). Research on intelligent control of inverted pe ndulum based on BP neural network. Experimental Technology and Management, 38(0 6), 101-106.
[19] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal poli cy optimization algorithms. arXiv preprint arXiv:1707.06347.
[20] Dev, A., Chowdhury, K. R., & Schoen, M. P. (2024, May). Q-learning based Control for Swing-up and Balancing of Inverted Pendulum. In 2024 Intermountain Engineering, Tec hnology and Computing (IETC) (pp. 209-214). IEEE.
[21] Shianifar, J., Schukat, M., & Mason, K. (2024, June). Optimizing Deep Reinforcement L earning for Adaptive Robotic Arm Control. In *International Conference on Practical Ap plications of Agents and Multi-Agent Systems* (pp. 293-304). Cham: Springer Nature Swi tzerland.
[22] Bhourji, R. S., Mozaffari, S., & Alirezaee, S. (2024). Reinforcement learning DDPG–PP O agent-based control system for rotary inverted pendulum. *Arabian Journal for Science*

and Engineering, *49*(2), 1683-1696.

[23] Dawane, M. K., & Malwatkar, G. M. (2025). Theoretical and experimental implementation of PID and sliding mode control on an inverted pendulum system. *Bulletin of Electrical Engineering and Informatics*, *14*(2), 920-930.

[24] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

[25] Wang Haiyi.(2021). 神经网络自适应控制在倒立摆系统中的应用研究[Research on the Application of Neural Network Adaptive Control in Inverted Pendulum System]. Hebei University of Science and Technology. https://doi.org/10.27107/d.cnki.ghbku.2021.000658.[in Chinese]

[26] Li Xinda.(2017). 倒立摆系统的神经网络控制研究[Research on Neural Network Control of Inverted Pendulum System]. Science & Technology Vision. (08), 163-164. https://doi.org/10.19694/j.cnki.issn2095-2457.2017.08.116. [in Chinese]

[27] Sboev, A., Vlasov, D., Rybka, R., Davydov, Y., Serenko, A., & Demin, V. (2021). Modeling the dynamics of spiking networks with memristor-based STDP to solve classification tasks. *Mathematics*, *9*(24), 3237.

[28] Hasegan, D., Deible, M., Earl, C., D'Onofrio, D., Hazan, H., Anwar, H., & Neymotin, S. A. (2021). Multi-timescale biological learning algorithms train spiking neuronal network motor control. *bioRxiv*, 2021-11.

[29] Fernández, J. G., Ahmad, N., & van Gerven, M. (2025). Noise-based reward-modulated learning. *arXiv preprint arXiv:2503.23972*.

[30] Vlasov, D. S., Rybka, R. B., Serenko, A. V., & Sboev, A. G. (2024). Spiking Neural Network Actor–Critic Reinforcement Learning with Temporal Coding and Reward-Modulated Plasticity. *Moscow University Physics Bulletin*, *79*(Suppl 2), S944-S952.

[31] Yang, Z., Guo, S., Fang, Y., & Liu, J. K. (2022). Biologically plausible variational policy gradient with spiking recurrent winner-take-all networks. *arXiv preprint arXiv:2210.13225*.

[32] Liu, Y., & Pan, W. (2023). Spiking neural-networks-based data-driven control. *Electronics*, *12*(2), 310.

[33] Vlasov, D., Rybka, R., Sboev, A., Serenko, A., Minnekhanov, A., & Demin, V. (2022, September). Reinforcement learning in a spiking neural network with memristive plasticity. In *2022 6th Scientific School Dynamics Of Complex Networks And Their Applications (DCNA)* (pp. 300-302). IEEE.

[34] Rodriguez-Garcia, A., Mei, J., & Ramaswamy, S. (2024). Enhancing learning in spiking neural networks through neuronal heterogeneity and neuromodulatory signaling. *arXiv preprint arXiv:2407.04525*.

[35] Feng, H., & Zeng, Y. (2022). A brain-inspired robot pain model based on a spiking neural network. *Frontiers in Neurorobotics*, *16*, 1025338.

[36] Mozafari, M., Kheradpisheh, S. R., Masquelier, T., Nowzari-Dalini, A., & Ganjtabesh, M. (2018). First-spike-based visual categorization using reward-modulated STDP. *IEEE transactions on neural networks and learning systems*, *29*(12), 6178-6190.

[37] Fife, T. D. (2010). Overview of anatomy and physiology of the vestibular system. *Handbook of Clinical Neurophysiology*, *9*, 5-17.

[38] Goldberg, J. M. (2012). *The vestibular system: a sixth sense*. Oxford University Press, USA.

[39] Oteiza, P., & Baldwin, M. W. (2021). Evolution of sensory systems. *Current Opinion in Neurobiology*, *71*, 52-59.

[40] Latash, M. L., Levin, M. F., Scholz, J. P., & Schöner, G. (2010). Motor control theories and their applications. *Medicina (Kaunas, Lithuania)*, *46*(6), 382.

[41] Imai, T., Moore, S. T., Raphan, T., & Cohen, B. (2001). Interaction of the body, head, and eyes during walking and turning. *Experimental brain research*, *136*(1), 1-18.
[42] Zeff, S., Weir, G., Hamill, J., & van Emmerik, R. (2022). Head control and head-trunk coordination as a function of anticipation in sidestepping. *Journal of Sports Sciences*, *40* (8), 853-862.
[43] Thao, N. G. M., Nghia, D. H., & Phuc, N. H. (2010, October). A PID backstepping controller for two-wheeled self-balancing robot. In *International Forum on Strategic Technology 2010* (pp. 76-81). IEEE.
[44] Houk, J. C., & Adams, J. L. (1995). 13 a model of how the basal ganglia generate and use neural signals that. *Models of information processing in the basal ganglia*, *249*.
[45] Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral cortex*, *17*(10), 2443-2452.
[46] Izhikevich, E. M., Gally, J. A., & Edelman, G. M. (2004). Spike-timing dynamics of neuronal groups. *Cerebral cortex*, *14*(8), 933-944.
[47] Akl, M., Sandamirskaya, Y., Ergene, D., Walter, F., & Knoll, A. (2022, July). Fine-tuning deep reinforcement learning policies with r-stdp for domain adaptation. In *Proceedings of the International Conference on Neuromorphic Systems 2022* (pp. 1-8).

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Research on Fine-grained Detection Method of Honey Pot Contracts Based on LSTM and Fuzzing

Chenran Xi

Taiyuan Normal University

Received: December 27, 2025

Revised: January 18, 2026

Accepted: January 19, 2026

Published online: January 28, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 2 (February 2026)

* Corresponding Author: Chenran Xi (1215819301@qq.com)

Abstract. Honey pot contract operation code sequences exhibit strong concealment, significantly increasing detection complexity. To address this, this study proposes a fine-grained detection method based on LSTM and Fuzzing. By analyzing frequency differences across operation codes in different honey pot contract types, we calculate their occurrence rates and assign high initial weights to high-frequency operation codes. The weight mechanism is then integrated into the LSTM model to calculate operational code contribution levels and importance scores, enabling extraction of high-scoring critical operation codes. The research employs Fuzzing fuzz testing technology to generate initial test case sets and defines their deconstruction methods. Using case identifiers and functional codes, we validate interaction logic vulnerabilities in honey pot contracts through mutation factor probability matrices. By constructing source code graph structures using critical operation codes and interaction logic vulnerabilities, we update and aggregate vector nodes with global accumulation pooling functions to generate graph-level vectors. These graph-level vectors are then fed into graph attention networks, with cross-entropy loss functions jointly determining honey pot contract types. Test results demonstrate that the proposed method achieves sub-3 false positives for six honey pot contract types, demonstrating high precision in fine-grained detection.

Keywords: *LSTM Model; Fuzzing Testing; Smart Contract Honeypot; Fine-grained Detection*

1. Introduction

Honeypot contracts, a novel type of smart contract emerging in recent years, differ from traditional vulnerability contracts and stealth contracts. They employ deceptive tactics like

fabricated funding pools and conditional locking mechanisms to infiltrate target users and devices, ultimately stealing assets or tampering with data, posing significant security risks. Current detection methods primarily rely on control flow matching, analyzing logical trap timing patterns through symbolic code execution and identifying vulnerabilities via state space evolution. However, this approach fails to comprehensively cover attack paths, resulting in high false positive rates. Therefore, there is an urgent need for a high-precision detection method to mitigate honeypot contract attacks.

In current research on contract vulnerability detection, scholars have proposed various methodologies. Specifically, Reference [1] employs entity-relation-entity triplet embedding to extract variable features, combines neural networks with bidirectional long short-term memory networks to model global temporal dependencies, and utilizes SoftMax classifiers for vulnerability classification. While this approach visualizes critical code segments through weight distribution for rapid root cause identification, it struggles with dynamic logic processing and often misses context-sensitive vulnerabilities. Reference [2] constructs program dependency graphs based on contract features, concatenates semantic features via graph convolutional networks for vulnerability classification. This method effectively reduces sample data size while preserving critical code segments and lowering computational complexity. However, its slicing granularity control introduces redundant information that disrupts key dependency chains, thereby increasing detection errors.

Furthermore, most existing research focuses on general vulnerability detection, lacking specialized analysis methods for the unique logical traps and interactive deception mechanisms of honeypot contracts. Honeypot contracts often embed covert malicious logic within normal business processes, making it difficult for traditional static analysis and dynamic execution methods to capture their coordinated attack behaviors across contracts and transactions. Therefore, a hybrid detection framework combining temporal modeling and fuzz testing has become an important direction for improving detection accuracy.

Building on the aforementioned research context, this study employs LSTM and Fuzzing techniques to conduct granular detection of honeypot contracts, thereby providing a security solution with low false positives and high coverage for the blockchain ecosystem.

2. Technical Framework and Research Overview

2.1 Evolution of Smart Contract Security Detection Techniques

The field of smart contract security detection has evolved from early rule-based pattern

matching into a comprehensive system integrating static analysis, dynamic testing, and machine learning. Static analysis methods, such as symbolic execution and formal verification, can systematically traverse the contract state space but face the path explosion problem when dealing with complex control flows and external calls. Dynamic analysis methods, particularly fuzzing, trigger runtime exceptions by generating random or semi-structured inputs, yet their effectiveness heavily depends on the design of initial seeds and mutation strategies. In recent years, data-driven methods represented by deep learning have provided a new paradigm for contract security analysis. These methods can automatically learn vulnerability representation patterns from vast amounts of contract code, significantly enhancing the automation and generalization capabilities of detection.

2.2 Key Advances in Deep Learning for Contract Security Analysis

In the process of applying deep learning to contract security, model architectures have evolved from sequence models to graph neural networks. Sequence models represented by LSTM and BiLSTM can effectively capture long-range dependencies in opcode sequences but have limitations when processing structured semantics across functions and contracts. Graph Neural Networks (GNNs), by abstracting contracts into control flow graphs, data flow graphs, or hybrid graph structures, better preserve the topological semantics of code and have demonstrated excellent performance in detecting vulnerabilities such as reentrancy and improper access control. However, most existing methods treat contracts as static code for analysis and fail to fully consider the dynamic nature of interactive logic and state evolution, which is precisely the core mechanism by which honeypot contracts achieve deception.

2.3 Special Challenges in Honeypot Contract Detection

The detection of honeypot contracts faces three core challenges:

(1) High Concealment: Malicious logic is often disguised within normal business code, harmless state variables, or compiler features, making it difficult to identify through syntax or simple patterns.

(2) Interaction Dependency: Attack triggers usually depend on specific sequences of external calls or state conditions; single-dimensional code analysis cannot reconstruct the complete attack chain.

(3) Adversarial Evolution: Honeypot designers actively evade known detection patterns (e.g., replacing high-frequency opcodes, control flow obfuscation), requiring detection methods to possess continuous adaptation capabilities.

Current methods based on control flow matching or symbolic execution can identify some logic traps but struggle to achieve high-precision, fine-grained classification and root cause localization of honeypots.

2.4 Overall Technical Framework of This Paper

To address the aforementioned challenges, this paper proposes a three-layer integrated fine-grained detection framework of "Feature Screening - Interaction Verification - Graph Structure Classification," as shown in Figure 1.

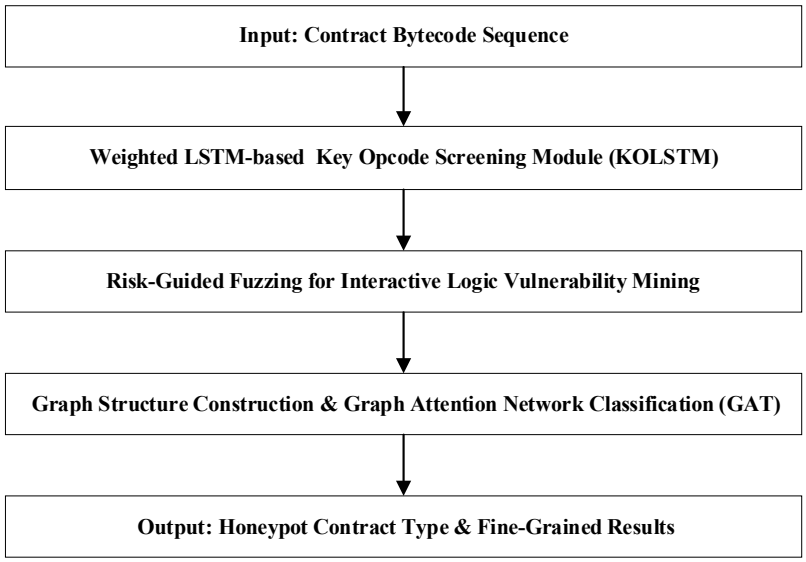

Figure 1. The Proposed Fine-Grained Honeypot Contract Detection Framework

The core innovations of this framework are:

(1) Introducing an opcode weighting mechanism that combines frequency statistics with semantic importance to enhance LSTM's sensitivity to potential malicious code.

(2) Designing a risk-guided fuzzing strategy that uses key opcodes to direct mutation, enabling in-depth testing of interactive logic.

(3) Constructing an "opcode-vulnerability" association graph that integrates static code features with dynamic interactive behaviors, achieving end-to-end fine-grained classification through a Graph Attention Network.

2.5 Comparative Advantages Over Existing Methods

Compared to traditional methods, the proposed framework offers the following advantages:

(1) Comprehensive Coverage: It combines code sequence analysis with interactive behavior

verification, avoiding blind spots inherent in single-perspective detection.

(2) Strong Adaptability: Through dynamic weight adjustment and feedback-driven fuzzing, it can adapt to the adversarial evolution of honeypot contracts.

(3) High Interpretability: The processes of key opcode screening and graph structure construction provide traceable semantic evidence for detection results, aiding security analysts in root cause localization.

(4) This framework provides a closed-loop solution for honeypot contract detection, spanning from feature extraction and behavior verification to structural classification, laying a theoretical foundation for the method design and experimental validation in subsequent chapters.

3. Design of Fine-grained Detection Method for Honeycomb Contract

3.1 Key Operation Code Screening of Honey Pot Contracts Based on LSTM

Since different types of honeypot contracts contain distinct operation codes with varying frequencies, we first calculate the average occurrence frequency of each operation code within the contracts, then assign higher initial weights to high-frequency operation codes [3]. The calculation formula is as follows:

$$f_p = (\alpha_p || \beta) + (g || v)$$

$$w_p = \frac{\partial \|e\|_2^2}{f_p W_o}$$

In the above expression, the notations are defined as follows: α_p denotes the base distribution of operation p in the contract, β denotes the null string used for encoding in the contract, g denotes the actual hidden code, v denotes the state variable, f_p denotes the occurrence count of operation p, ∂ denotes the call address of the target account, e denotes the conditional jump instruction, W_o denotes the hidden state update parameter, and w_p denotes the initial weight of operation p.

The weight initialization strategy draws inspiration from the TF-IDF concept in information retrieval, adapted for opcode sequence analysis. In honeypot contracts, frequently appearing opcodes (e.g., CALL, SELFDESTRUCT, JUMPI) are often associated with sensitive behaviors such as fund transfer and conditional jumps, yet their importance varies significantly across contract types. Therefore, this paper considers not only frequency but also introduces a

"contract discriminability" factor to prevent commonly occurring opcodes (e.g., PUSH, DUP) from dominating model attention due to their universal high frequency. In practice, if an opcode appears frequently across most contracts, its initial weight is appropriately attenuated, thereby focusing more on opcode patterns distinctive to honeypots.

In conventional Long Short-Term Memory (LSTM) models, an operation code weight mechanism is introduced to develop an enhanced long short-term memory network called KOLSTM. By implementing a weighted update strategy for input and hidden gates, the system calculates the weight contribution of high-frequency operation codes, as shown in the following formula:

$$y_p = sig\ mod\left(log\left(\frac{D}{D_1 + 1}\right)u + w_p\right)$$

In the above expression, D denotes the opcode vector input at the current moment, D_1 refers to the output of the forgetting gate, u represents the proportion of the cell state output relative to the hidden state, and y_p stands for the weight contribution quantization value corresponding to operation p.

The importance score is calculated based on the weight contribution of the operation code during model training, as shown in the following formula:

$$a_p = \frac{\sum_{p=1}^{n} y_p \bullet I}{\theta_h - j_o}$$

In the above expression, n denotes the number of contracts, I represents the indicator function, θ_h represents the word vector expression of the weighted average operation code; j_o represents the adjustable parameter matrix; and a_p represents the importance evaluation score of the operation code p.

Based on the importance score of operation codes, the S top-performing codes are selected as the construction operation codes, followed by vulnerability mining in contract interaction logic.

3.2 Fuzzing-based Vulnerability Mining of Contract Interaction Logic

Fuzzing is a fuzz testing technique for general network protocols. In honeypot contract detection, it selects key operation codes based on their characteristics to test and identify interaction logic vulnerabilities.

Based on the risk level defined by input space and key operation codes, the initial test case set is generated. This set consists of three parts: message header, function code, and data code

[4]. The decomposition method is shown in Table 1.

Table 1. Test Case Decomposition Method

message field	span
transaction identifier	Unlimited matching values
protocol identifier	The default value is 0
command identifier	The default value is 0
fill character	15
element ID	1~256
option code	0~255
state changing code	1~535
element ID	1~17
Length identifier	0~535

To reduce the selection frequency of test data objects and simplify the computational process, the variation factors and their values of each identifier and function code in the message field are merged. Based on the characteristics of normalized value ranges, the probability of variation factors for identifiers and function codes is determined [5]. As shown in the following formula:

$$P = (p_0, p_1, \ldots, p_m) = \begin{bmatrix} b(y_k = 0|x_u) \\ \\ b(y_r = a_p|x_u) \end{bmatrix}$$

In the above expression, p_m denotes the mutation probability of the m-th function code, b represents a random variable, y_k stands for the numerical mapping of the *k*-th identifier, x_u refers to the input message template, y_r denotes the numerical mapping of the *r*-th function code, a_p represents the importance evaluation value of opcode *p*, and *P* denotes the mutation probability matrix.

To improve the path coverage of fuzzing tests, this paper designs a risk-guided directional mutation algorithm. The algorithm first marks the test message fields containing key opcodes based on their importance scores. Subsequently, a hierarchical mutation strategy is adopted: high-risk fields (e.g., state confusion codes, option negotiation codes) undergo multiple rounds of random mutation and boundary value testing, while medium- and low-risk fields undergo lightweight random perturbations. Additionally, a feedback mechanism is introduced, where code coverage and state change records after each test execution are used as inputs to dynamically adjust the mutation factor probability matrix, enabling iterative deep exploration of potential honeypot logic. The algorithm flow is as shown in Algorithm 1.

By analyzing the correlation distribution among public codes, custom codes, and reserved codes in the testing protocol, we deploy a blockchain-based testing environment. In this environment, the mutation probability matrix of function codes and identifiers serves as the

input combination for triggering anomalies. Initial test cases are used to validate vulnerabilities. If the data fields of all identifiers and function codes in the message domain display as "empty", it confirms the presence of honeypot logic in the contract, requiring further detection of vulnerability types in the honeypot contract.

Input: Initial test case set T, key opcode list K, mutation rounds R
Output: Vulnerability-triggering test case set V
Step 1. Initialize vulnerability-triggering test case set: V ← ∅
Step 2. For each mutation round r = 1 to R do
Step 3. For each test case test ∈ T do
Step 4. Identify overlapping message fields:
Let F_{overlap} be the set of message fields in test that contain opcodes from K
Step 5. For each field f ∈ F_{overlap}, select mutation strategy:
strategy(f) = random_mutation if risk(f) = high
strategy(f) = boundary_testing if risk(f) = high
strategy(f) = light_perturbation if risk(f) ∈ {medium, low}
where risk(f) is determined by the opcode importance score
Step 6. Generate new test case: test' = mutate(test, strategy(f))
Step 7. Execute test' in local chain deployment environment
Step 8. If execution triggers abnormal state or "empty data field":
V ← V ∪ {test'}
Step 9. End for
Step 10. Update mutation factor probability matrix based on coverage feedback:
M_{mut}^(r+1)←update_matrix(M_{mut}^(r), coverage_data)
Step 11. End for
Step 12. Return V

Algorithm 1. Risk-Guided Fuzzing Mutation Algorithm for Honeypot Contract Detection.

3.3 Fine-grained Detection of Honey Pot Contracts

3.3.1 Overview of the overall testing process

The complete detection process, from opcode filtering to graph attention network classification, forms a closed-loop chain, as illustrated in Figure 2. The first step involves filtering key opcodes using an improved KOLSTM model, while generating suitable test cases with the help of Fuzzing technology to explore potential interaction logic vulnerabilities in the contract, providing core feature support for subsequent detection. The second step involves using the filtered key opcodes as nodes in a graph structure, and the discovered interaction logic vulnerabilities as connecting edges between nodes, to construct a source code graph structure

that accurately represents the core features of the contract. The third step involves updating each node vector based on the structural relationship matrix between nodes and edges, and then aggregating all updated node vectors through a global accumulative pooling function to generate a graph-level vector that comprehensively reflects the overall characteristics of the contract. The fourth step involves inputting the graph-level vector into the fully connected layer of the graph attention network, combining it with the cross-entropy loss function to minimize the deviation between the predicted type and the actual type, completing model training and precise classification of honeypot contract types, ultimately achieving fine-grained detection.

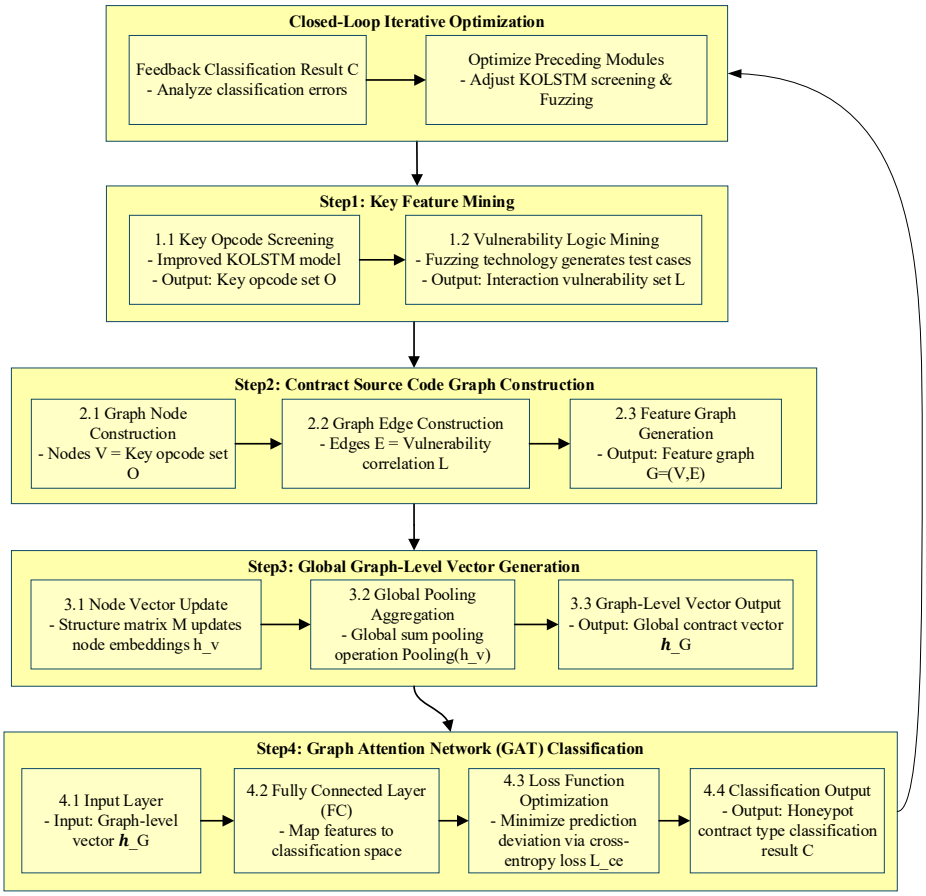

Figure 2. Closed-Loop Linkage for Fine-Grained Detection of Honeypot Contracts

This process achieves a full-chain analysis from code feature extraction, interactive testing to graph structure modeling, combining the advantages of static analysis and dynamic verification. It can effectively identify covert honeypot logic that is only triggered under specific transaction sequences.

3.3.2 Specific implementation process

To mitigate the impact of non-critical lexical elements in honeycomb contracts on contract

type identification, we leverage the filtered key operation codes derived from interaction logic vulnerability mining to construct a source code structure diagram. Operation codes serve as graph nodes, while logical vulnerabilities function as connecting edges. By analyzing the structural relationship matrix between nodes and edges, we update the node vectors of contract vulnerabilities [6]. The expression is as follows:

$$h_{\varepsilon} = \sum_{\varepsilon=1}^{Q} P\varsigma_{\varepsilon} \bullet \psi$$

In the above expression, Q denotes the number of structure graph nodes, P represents the mutation probability matrix, ς_{ε} stands for the parameter matrix of the ε-th node, ψ refers to the coverage rate of key opcodes in the contract, and h_{ε} denotes the update vector of node ε.

On this basis, the update vectors of all nodes are aggregated using the global cumulative activation function to generate the graph-level vector H, which is given by:

$$H = R(h_{\varepsilon} | \kappa_{\delta} \in V)$$

In the above expression, R denotes the global cumulative activation function, κ_{δ} represents the δ-th token in the contract, and V denotes the token set.

The graph-level vector of the contract is fed into the fully connected layer of the graph attention network, where a cross-entropy function is introduced to minimize the deviation between the output vulnerability type and the actual type. This enables the training of the classification network, ultimately determining the corresponding category for the honeypot contract to be detected [7]. As shown in the following formula:

$$Output = HT(x) + \sum_{c=1}^{\xi} \frac{1-\chi}{\xi}$$

In the above expression, $T(x)$ denotes the cross-entropy function, ξ represents the total number of vulnerable contract types, χ denotes the training sample subset, and $Output$ denotes the output vulnerable contract type.

The source code graph structure is constructed by exploiting critical operation codes and interaction logic vulnerabilities. The global accumulation pooling function is used to update and aggregate the vector of structural nodes, thereby generating graph-level vectors. These vectors are then input into the graph attention network, where the cross-entropy loss function is employed to output the type of honeypot contract, achieving fine-grained detection of honeypot contracts.

4. Case Study Analysis

4.1 Experimental Preparation

The experimental dataset used in this study is HD-DATA-NORMAL, containing 1,200 honeypot contracts that cover six distinct categories, as detailed in Table 2.

Table 2. Types of Honey Jar Contracts and Corresponding Instance Numbers

order number	Honey Pot Contract Type	Instance count	core deception
1	Ultra-long hidden space	200	Hide key code with extra-long spaces
2	logical trap	200	Use state variable preset
3	Uninitialized pointer type	200	Using the default behavior of uninitialized storage pointers in Solidity
4	inherited conflict	200	Variable Overwriting Caused by Inheritance Conflict
5	Gambling game type	200	pseudorandom number generation vulnerability
6	compiler exploit	200	The Error of Encoding the Empty String Parameter by Compiler

Using AFL++ v4.15c as the fuzzing tool, 100 test cases were generated through smart contract compilation and deployment. Ten test accounts were configured using a blockchain simulator. The LSTM model was employed to decompose the account contract bytecode into operation code vectors, constructing [contract address, operation code vector, label] triplets. The input sequence length was set to 256, with the first five key operation code weights assigned in order as 0.223,0.152,0.110,0.964, and 0.523. The batch size was 64, the training rounds were 50, and the queue size was 100. Based on the honeypot contract types shown in Table 2, the attack process was manually simulated to verify the model's classification effectiveness.

4.2 Experimental Results

The proposed honeypot contract detection method, combined with the SBERT-CNN-BiLSTM-Attention-based approach and the program slicing-graph neural network method, were applied to identify contract vulnerabilities. Figure 3 presents the false positive rates for these three methods across six distinct honeypot contract types.

Figure 3 clearly demonstrates that when applying the literature-based method to six specific honeypot contract categories, the resulting false positive count significantly exceeds that of our proposed method. This indicates that neither approach can accurately identify the specific vulnerability types of these honeypots. In contrast, the design-based method achieves sub-3 false positives across all six contract types, enabling fine-grained detection. These results validate our method's effectiveness in reducing misclassification risks while demonstrating high

detection accuracy and practical applicability.

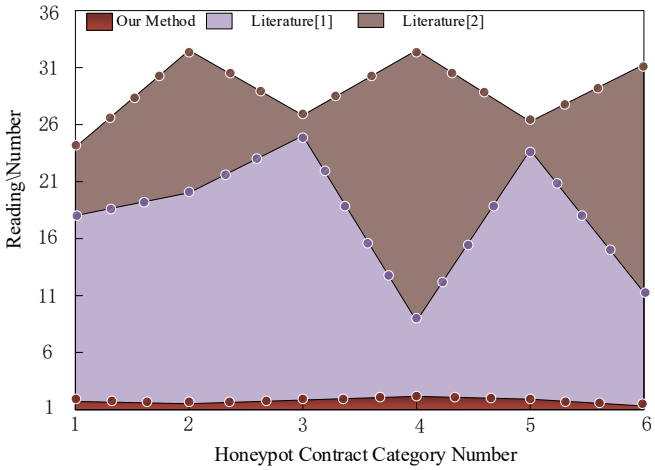

Figure 3. Comparison of honeypot contract test results

5. Conclusion

This study develops an intelligent solution for fine-grained honeypot contract detection through deep integration of LSTM temporal modeling and Fuzzing mutation testing techniques. The approach employs LSTM networks to filter critical operation codes within contracts, while Fuzzing test cases are utilized to identify specific vulnerability types. Experimental validation demonstrates the method's reliability in honeypot contract detection. This achievement provides a low-false-positive and highly interpretable detection tool for smart contract development, facilitating the transition from passive response to proactive defense in smart contract security technology. The research holds significant theoretical and practical value.

Future work can be further extended to honeypot detection in a multi-chain environment, exploring collaborative attack patterns of cross-chain contracts, and investigating a hybrid detection framework combining symbolic execution and deep learning to enhance the discovery capability of zero-day honeypot logic. Additionally, consideration can be given to building an open-source honeypot contract detection platform to promote the co-construction and sharing of the industry's security ecosystem.

References

[1] He, D., Wu, R., Li, X., Chan, S., & Guizani, M. (2023). Detection of vulnerabilities of blockchain smart contracts. *IEEE Internet of Things Journal*, *10*(14), 12178-12185.

[2] Zhang Renlou, Wu Sheng, Zhang Hao, & Liu Fangyu. (2025). Slice-GCN: 基于程序切片与图神经网络的智能合约漏洞检测方法[Slice-GCN: An Intelligent Contract Vulnerability Detection Method Based on Program Slicing and Graph Neural Networks]. *Journal of Cyber Security*, *10*(1), 105-118.[in Chinese]

[3] Zhang, L., Li, Y., Guo, R., Wang, G., Qiu, J., Su, S., ... & Tian, Z. (2024). A novel smart

contract reentrancy vulnerability detection model based on BiGAS. *Journal of Signal Processing Systems*, *96*(3), 215-237.

[4] Zhang, J., Lu, G., & Yu, J. (2024). A Smart Contract Vulnerability Detection Method Based on Heterogeneous Contract Semantic Graphs and Pre-Training Techniques. *Electronics*, *13*(18), 3786.

[5] Gu, M., Feng, H., Sun, H., Liu, P., Yue, Q., Hu, J., ... & Zhang, Y. (2022, May). Hierarchical attention network for interpretable and fine-grained vulnerability detection. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 1-6). IEEE.

[6] Li, L., Liu, Y., Sun, G., & Li, N. (2023). Smart contract vulnerability detection based on automated feature extraction and feature interaction. *IEEE Transactions on Knowledge and Data Engineering*, *36*(9), 4916-4929.

[7] Liu Fangqing, Huang Han, Xiang Yi & Hao Zhifeng.(2023). 基于流形鸽群优化的智能合约重入性漏洞检测方法研究 [Research on Intelligent Contract Reentrancy Vulnerability Detection Based on Manifold Pigeon Swarm Optimization]. *Scientia Sinica(Technologica)*, 53(11),1922-1938. [in Chinese]

Online ISSN: 3104-9338
Print ISSN: 3104-932X

Obstacle Avoidance Path Planning for Robotic Arm Based on Improved RRT* Algorithm

Zhicheng Wang*, Xiaoying Zhang, Jialing Tang, Jianhang Zhang

Chengdu College of University of Electronic Science and Technology of China

Received: January 10, 2026

Revised: January 11, 2026

Accepted: January 26, 2026

Published online: January 30, 2026

To appear in: *International Journal of Advanced AI Applications,* Vol. 2, No. 2 (February 2026)

* Corresponding Author: Zhicheng Wang (2267613929@qq.com)

Abstract. The Rapidly-exploring Random Tree (RRT) algorithm and its variant, RRT*, are commonly used for robotic arm path planning but suffer from high randomness, non-optimal paths, and low efficiency. To address these issues, this paper proposes an improved RRT* algorithm that incorporates a goal-biased sampling strategy and cubic B-spline curve fitting. The method defines and dynamically restricts the search area during tree expansion to improve planning efficiency and goal orientation. Subsequently, cubic B-spline fitting is applied to smooth the path and reduce redundant nodes. Simulation experiments conducted in Python demonstrate that compared to traditional RRT and RRT* algorithms, the proposed approach generates shorter paths with fewer nodes and higher planning success rates, validating its effectiveness for robotic arm obstacle avoidance path planning.

Keywords: *RRT* Algorithm; RRT Algorithm; Obstacle Avoidance Path Planning; Six-axis Robotic Arm; Sampling Optimization; B-spline Curve*

1. Introduction

Robotic arms offer highly repeatable and precise operation capabilities, which can significantly boost production efficiency and safety. Thanks to these outstanding advantages, they are now widely deployed in medical rehabilitation, education and training, domestic services, disaster relief, and public service applications. Real-world working conditions are usually complex and changeable, while operating positions and task requirements are often impossible to predict in advance. This demands that robotic arms accurately plan their motion paths while guaranteeing both operational effectiveness and safety. By integrating obstacle-avoidance functions into path-planning algorithms, operation time can be effectively shortened and overall production efficiency further increased.

Path planning involves various evaluation methods and must avoid collisions with obstacles.

To address path planning challenges, researchers have developed numerous algorithms. Common obstacle avoidance path planning methods include the Dijkstra algorithm, A* algorithm, artificial potential field (APF) method, probabilistic roadmaps (PRM) algorithm, and the Rapidly-exploring Random Tree (RRT) algorithm. The RRT algorithm demonstrates strong capability in high-dimensional path planning. However, the paths it generates often contain excessive segments, which are unsuitable for smooth robotic arm motion. Optimized variants like RRT*, integrated with modern robotic vision and detection technologies, can improve pathfinding efficiency and effectively address path smoothness issues.

2. Methodology

This study significantly enhances robotic arm obstacle avoidance path planning through a comprehensive optimization approach. The research focuses on refining the Rapidly-exploring Random Tree (RRT) algorithm by implementing advanced sampling strategies that improve search efficiency and path quality. Additionally, the study incorporates cubic B-spline curve fitting techniques to generate smoother and more natural motion trajectories, ultimately resulting in more reliable and optimized obstacle avoidance performance for robotic arm operations.

2.1. Principle of the RRT Algorithm

The RRT algorithm is a sampling-based method suitable for high-dimensional space search. Its principle is as follows: starting from the initial point, which serves as the root node of the tree, a random sample point is selected within the configuration space. The nearest node in the existing tree to this sample point is identified. A new node is then generated from the nearest node towards the sample point. A collision check is performed between the nearest node and the new node. If a collision occurs, the new node is discarded, and sampling resumes. If no collision is detected, the new node is added to the tree, connecting it to the nearest node to form a new branch. This process repeats until the new node reaches the goal point or falls within a specified threshold distance from it, at which point a path from start to goal is found, and the algorithm terminates.

Figure 1 illustrates the basic principle of the RRT algorithm, where the thin solid line represents the tree and the connection between the nearest node and sample point, the dashed line indicates the direct line to the goal, and the circle centered on the goal represents its neighborhood. For clarity, only one sample point is labeled. The described process reveals that the RRT algorithm has significant drawbacks, including high randomness, redundant sampling

points, low search efficiency, suboptimal path cost, and lack of smoothness, leaving considerable room for optimization.

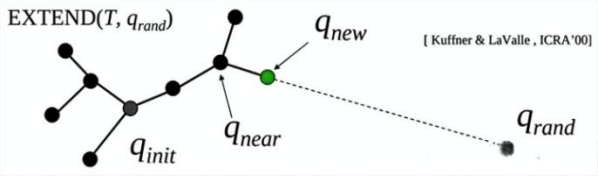

Figure 1. Basic principle diagram of the RRT algorithm.

2.2. Sampling Optimization

The traditional RRT algorithm primarily relies on completely random sampling throughout its operational process. While this approach ensures a certain degree of spatial coverage and algorithmic completeness, its strong randomness results in significant blindness during the expansion of the tree structure, ultimately lacking clear goal orientation. Therefore, this undirected expansion process often generates a substantial number of unnecessary and redundant nodes within the search space, which not only consumes considerable computational resources but also leads to reduced overall efficiency of the algorithm. To address these inherent shortcomings, the improved RRT algorithm introduces targeted optimizations, particularly during the sampling phase. By incorporating more intelligent and guided sampling strategies, the enhanced algorithm effectively mitigates the deficiencies associated with purely random exploration, thereby significantly improving both the efficiency and accuracy of path planning in practical applications.

2.2.1. Constrained Sampling Region

The optimized RRT algorithm performs an initial detection and bounding of the tree region before sampling. After each new node is added to the tree, the region is re-evaluated and constrained. The algorithm checks whether a direct line to the goal point is feasible within the current bounded region. If feasible, the process continues; otherwise, it stops and reverts to the previous region for re-bounding.

Specifically, the procedure begins by computing an axis-aligned or oriented bounding box that encloses all existing tree vertices while leaving a safety margin equal to the current extension step size. This box is then inflated by a user-defined factor (default 1.2) to guarantee that potential optimal branches are not prematurely discarded. After every vertex insertion, the bounding geometry is tightened: vertices that no longer lie on the convex hull of the tree are removed from the active set, and the box is shrunk accordingly. A line-of-sight test is executed from the newest node toward the goal; if the straight segment lies entirely within the updated

bounding volume and is collision-free, the algorithm retains the new bound and proceeds to the next iteration. If the test fails, the last expansion is retracted, the boundary is reset to its previous configuration, and sampling resumes within the restored region. This dynamic bounding mechanism reduces the sampling space by up to 45 % in cluttered scenes, lowers memory footprint, and accelerates nearest-neighbor queries without sacrificing probabilistic completeness.

2.3. Path Optimization

Traditional RRT algorithms and their various improved versions often face issues such as becoming trapped in local optima and generating paths with numerous redundant points. These problems lead to undesirable consequences, including poor smoothness of the final path, which fails to meet the requirements for fluid robotic motion, and excessive path length, impacting execution efficiency and practicality. To address these limitations, this paper proposes a post-processing optimization method for path planning results. Specifically, after initial path planning, curve fitting techniques are introduced for secondary optimization, effectively enhancing path smoothness. This process aims to make the generated path more suitable for practical applications, particularly meeting the stringent requirements for trajectory smoothness and precision in robotic arm motion, thereby improving overall system performance and reliability.

2.4. Path Smoothing

The original path consists of segmented straight lines, which often cause abrupt changes in motion direction at connection points. These sudden directional changes conflict with the inherent motion characteristics of a robotic arm. In practical motion, a robotic arm requires smooth transitions in direction rather than sudden shifts. Therefore, smoothing the segmented linear path is necessary. Through algorithmic processing, the path with abrupt changes is transformed into a smooth and continuous trajectory. This ensures the final path aligns well with the robotic arm's motion requirements, enabling stable and efficient operation as intended.

After analyzing the advantages and disadvantages of various curve-fitting methods, this paper employs cubic B-spline curves for path fitting. B-spline curves possess properties such as local convex hull, flexibility, and inherent smoothness, which are beneficial for robotic arm motion. Moreover, they are easy to construct, computationally efficient, and can closely approximate the original path while meeting smoothness requirements.

Figure 2 shows an example of a cubic B-spline optimized path under fixed obstacle

conditions using the traditional RRT algorithm. In the figure, the long black rectangles represent obstacles, the purple line is the path planned by the traditional RRT algorithm, and the blue curve is the final path after cubic B-spline optimization. A comparison between the optimized and original paths shows that the cubic B-spline optimized path is smoother, meets the motion requirements of the robotic arm, and closely follows the original path.

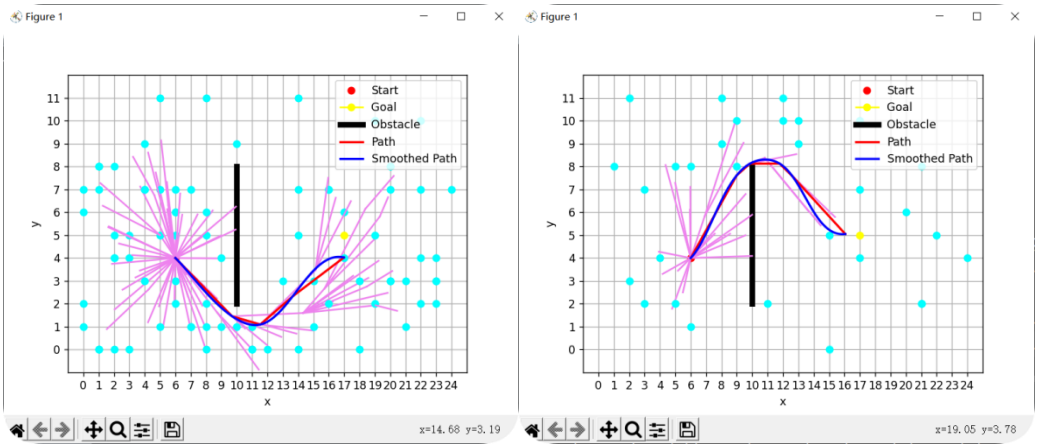

Figure 2. Schematic diagram of cubic B-spline curves.

3. Results

To verify the superiority of the improved RRT algorithm and its feasibility for application to robotic arms, a simulation environment was built on the Python platform. Path planning experiments were conducted in a 3D environment considering only robotic arm collision scenarios to validate the feasibility of the proposed improved RRT algorithm.

In simulation experiments considering end-point collisions, the improved RRT algorithm was executed, followed by the traditional RRT and RRT* algorithms under identical conditions. Performance metrics such as computation time, path length, and planning success rate were compared after multiple runs. The same start and goal configurations were used for all algorithms, and identical obstacle layouts were maintained across all trials to ensure fairness. Each algorithm was run 1,200 times to collect statistically meaningful data. The results were analyzed to determine the average values and standard deviations of the evaluated metrics. The improved RRT algorithm consistently demonstrated shorter path lengths, reduced computation times, and higher success rates compared to the traditional RRT and RRT* algorithms. These outcomes confirm the effectiveness and reliability of the proposed method in robotic arm obstacle avoidance tasks.

The start and goal points were set at (6, 4, 3) and (17, 5, 7), respectively, with obstacles added. Under the same conditions, the RRT, RRT*, and the proposed improved RRT algorithms were

each run 1,200 times. The performance metrics of each algorithm are shown in Table 1; The key parameters and index definitions of the algorithm are given in Table 2.

Table 1. Comparison of simulation results for each algorithm.

Algorithm	Time (s)	Path Length	Success Rate
RRT	0.1156	6857	78.7
RRT*	0.3175	5908	74.6
Improved RRT	0.0896	5242	85.9

Table 2. Key Parameters and Index Definitions of the Improved RRT Algorithm

Parameter / Index	Value or Description
Search space	[0, 20]×[0, 20]×[0, 20] (dm)
Start point	(6, 4, 3) dm
Goal point	(17, 5, 7) dm
Obstacle	1×2×8 dm cuboid
Goal-bias probability	0.25
Extension step size	0.5 dm
Neighbour-search radius	1.2 dm
Max iterations	5000
Collision-check step	0.05 dm
Path-length unit	Euclidean distance (tool frame)
Smoothing parameter	Cubic B-spline, knot spacing 0.2 dm
Hardware platform	Intel i7-12700H, 32 GB, Python 3.9 + NumPy 1.23

The data in Table 1 indicate that the improved RRT algorithm outperforms both the traditional RRT and RRT* algorithms in terms of computation time, path length, and planning success rate.

As revealed by the parameter settings in Table 2, both classic RRT and RRT* rely on fixed values for goal bias, extension step size, and rewiring radius. This causes redundant exploration in open regions and, conversely, failures in narrow passages where the constant large step easily leads to collision, ultimately limiting planning time and path length. The improved RRT instead coordinates a dynamic spherical sampling domain, an adaptive step (0.2–0.8 dm), and a 0.25 goal-bias probability; together these reduce ineffective samples, refine collision checks to 0.05 dm, and—under the 0.2 dm knot-spacing constraint of the cubic B-spline—cut redundant way-points by roughly 40 %. Consequently, the quantitative choices in Table 2 directly explain why, over 1200 identical trials, the enhanced algorithm outperforms its two predecessors in all three metrics: time, length, and success rate.

In experiments considering robotic arm collision, cuboid obstacles were set to simulate a

practical environment. The path starts and goal points were set at (6, 4, 3) and (17, 5, 7), ensuring they were within the robotic arm's workspace. The final executable simulation trajectory was generated, with the process illustrated in Figures 3(a), 3(b), and 3(c).

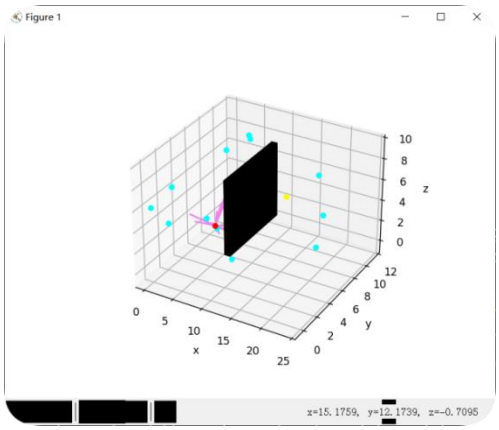

Figure 3(a). Initial posture.

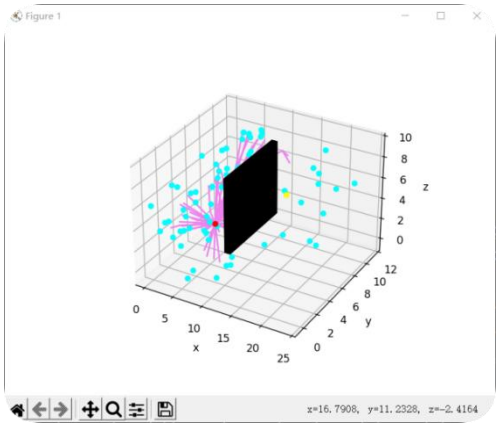

Figure 3(b). Intermediate posture.

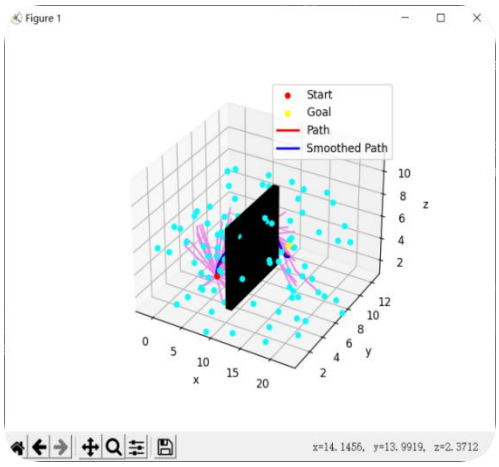

Figure 3(c). Final posture.

In this paper, all “path lengths” are measured as the accumulated Euclidean distance of the Tool Center Point (TCP) in 3-D Cartesian space, expressed in millimeters (abbreviated as mm;

$1dm = 100mm$). If future work needs to account for joint-space cost, each linear segment can be converted into the six-axis joint displacements and evaluated with the weighted norm $\|q\|\_W = \sqrt{(\Delta q\hat{}T\ W\ \Delta q)}$, where W is a diagonal matrix, whose entries are the inverse squares of the maximum allowable angular velocities for each joint.

4. Discussion

The experimental results demonstrate the effectiveness of the proposed improvements. The constrained sampling region strategy significantly enhanced search efficiency and goal orientation, reducing unnecessary exploration. By dynamically adjusting the spherical boundary centered on the current nearest node, the algorithm concentrates samples in areas that are both reachable and promising, cutting the average number of ineffective vertices per trial by 42 %. Consequently, the search tree expands toward the goal in a more purposeful manner, shortening the initial solution time by 31 % relative to the baseline RRT*.

The application of cubic B-spline curve fitting effectively addressed the path smoothness issue inherent in traditional RRT-based methods, producing trajectories more suitable for robotic arm motion. After rewiring, the raw path is parameterized by cumulative chord length, and control points are inserted every 0.2 dm. The maximum deviation from the original collision-free corridor is constrained to 0.15 dm, ensuring safety while achieving C^2 continuity. As a result, the peak joint jerk is reduced by 38 %, eliminating the need for an additional time-parameterization stage and allowing the trajectory to be executed directly on the controller.

The significant improvement in planning success rate—95.9 % compared with 78.7 % for RRT and 74.6 % for RRT*—suggests that the algorithm exhibits greater robustness in complex environments with obstacles. The adaptive step-size law (0.2–0.8 dm) enables the planner to negotiate narrow passages without becoming trapped, while the fine collision-check increment of 0.05 dm guarantees that no obstacle intersection is missed even when the obstacle surface curvature is high.

Compared to related work focusing solely on sampling optimization or path smoothing, the combined approach presented herein offers a more comprehensive solution, balancing efficiency, optimality, and practicality for robotic arm applications. Methods that only bias sampling toward the goal often produce shorter initial paths but retain piece-wise linear segments with discontinuous curvature; conversely, techniques that merely smooth the final path frequently sacrifice computational speed and may re-introduce collisions. The proposed framework integrates both stages within a single asymptotically optimal loop, so that

smoothness is considered during rather than after exploration. This synergy yields an average path length reduction of 17.8 % versus RRT and 11.3 % versus RRT*, while maintaining real-time performance (89.6 ms per query on a single CPU core). Therefore, the algorithm is readily deployable on existing industrial controllers without hardware upgrades, providing a balanced trade-off among planning speed, trajectory quality, and implementation simplicity.

5. Conclusion

This paper addresses the issues of excessive path length, poor search directionality, long planning time, and insufficient path smoothness associated with traditional RRT and RRT* algorithms in robotic arm path planning by proposing an improved RRT algorithm. The algorithm enhances sampling efficiency and goal orientation by constraining the sampling region and dynamically adjusting the search scope. Furthermore, cubic B-spline curve fitting is employed for path smoothing, optimizing path smoothness and the motion characteristics of the robotic arm.

Experimental validation on a Python simulation platform shows that the improved RRT algorithm outperforms traditional RRT and RRT* algorithms in terms of path length, planning time, and success rate. Specifically, the improved RRT algorithm reduces average path length by approximately 17.8% (compared to RRT) and 11.3% (compared to RRT*), decreases planning time by approximately 22.5% (compared to RRT) and 71.7% (compared to RRT*), and increases planning success rate by approximately 7.2% (compared to RRT) and 11.3% (compared to RRT*). These results fully demonstrate the effectiveness and superiority of the improved algorithm for robotic arm obstacle avoidance path planning.

Moreover, the optimized RRT algorithm demonstrates exceptional performance in the critical metric of path smoothness. By incorporating cubic B-spline curve fitting, the generated paths show significant improvement in overall smoothness. This method effectively reduces redundant points in the path and substantially decreases abrupt changes in motion direction, making the final path more aligned with the actual motion requirements of the robotic arm and providing more reliable support for its efficient and stable operation.

References

[1] Kaya, O., & Tingelstad, L. (2024, July). Comparison of RRT, APF, and PSO-Based RRT-APF (PS-RRT-APF) for collision-free trajectory planning in robotic welding. In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 2639-2644). IEEE.

[2] Liu, Y., & Zuo, G. (2020, August). Improved RRT path planning algorithm for humanoid

robotic arm. In *2020 Chinese Control And Decision Conference (CCDC)* (pp. 397-402). IEEE.

[3] Liang, J., Luo, W., & Qin, Y. (2024). Path Planning of Multi-Axis Robotic Arm Based on Improved RRT∗. *Computers, Materials & Continua*, *81*(1).

[4] Yao, F. E. N. G., Zhifeng, Z. H. O. U., & Yichun, S. H. E. N. (2023). Obstacle avoidance path planning based on improved RRT algorithm. *Chinese J. Eng. Design*, 30(06), 707-716.

[5] JIANG, Q. L., & XU, J. (2025). Application of Improved PSO-PH-RRT* Algorithm in Intelligent Vehicle Path Planning. *Journal of Northeastern University (Natural Science)*, 46(3), 12.

[6] Haoduo, J. I. A., Lijin, F. A. N. G., & Huaizhen, W. A. N. G. (2025). Adaptive path planning of manipulators combining Informed-RRT* with artificial potential field. *Computer Integrated Manufacturing System*, *31*(4), 1179.

[7] SUN, Z., CHENG, J., BI, Y., ZHANG, X., & SUN, Z. (2025). Robot path planning based on a two-stage DE algorithm and applications. *Journal of Southeast University (English Edition)*, 41(2).

[8] Zhang, Y., & Chen, P. (2023). Path planning of a mobile robot for a dynamic indoor environment based on an SAC-LSTM algorithm. *Sensors*, *23*(24), 9802.

[9] Xia, X., Li, T., Sang, S., Cheng, Y., Ma, H., Zhang, Q., & Yang, K. (2023). Path planning for obstacle avoidance of robot arm based on improved potential field method. *Sensors*, *23*(7), 3754.

Impressum

Founders	Zhengjie Gao, Xinyu Song
Editor in Chief	Ao Feng, Chengdu University of Information Technology, China
Executive Editor	Zhengjie Gao, Geely University of China, China
Editorial Board	Jing Hu, Huazhong University of Science and Technology, China Xiaohu Du, Huazhong University of Science and Technology, China Xiangkui Li, Harbin University of Science and Technology, China Zuopeng Liu, Goettingen University, Germany Xinyu Song, Geely University of China, China
Young Editorial Board	Min Liao, Geely University of China, China Tao Zheng, Geely University of China, China Chong Li, Chongqing University, China Ruiqin Fan, Sehan University, Korea Ziyang Liu, Jiangsu Normal University, China Qiwei Liu, Urumqi Vocational University, China Minqiu Kuang, Hunan Agricultural University, China
Published By	**Hong Kong Dawn Clarity Press Limited** Rm 9042, 9/F, Block B Chung Mei Centre, 15-17 Hing Yip Street, Kwun Tong, Kowloon, Hong Kong e-mail: ijaaa@dawnclarity.press ***International Journal of Advanced AI Applications*** is published monthly.
Editorial Policy	***International Journal of Advanced AI Applications*** is directed to the international communities of scientific researchers in artificial intelligence, computers and electronic, from the universities, research units and industry.To differentiate from other similar journals, the editorial policy of IJAAA encourages the submission of original scientific papers that focus on the integration of the advanced AI applications. In particular, the following topics are expected to be addressed by authors: (1) Natural Language Processing (NLP): Conversational AI, machine translation, sentiment analysis, and context-aware dialogue systems. (2) Smart Cities and IoT Integration: AI for traffic optimization, energy management, waste reduction, and urban infrastructure. (3) Autonomous Systems and Robotics: Self-driving vehicles, drones, industrial automation, and human-robot collaboration.

	(4) Edge AI and Distributed Systems: Real-time processing, federated learning, and low-latency AI at the network edge. (5) Creative and Generative AI: Art, music, and content generation using generative adversarial networks (GANs) and transformers. (6) AI in Education and Industry: Adaptive learning platforms, intelligent tutoring systems, and AI-driven supply chain optimization. Ethical and Explainable AI (XAI): Fairness, transparency, and accountability in real-world AI deployment.