ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Optimization method of robot navigation state space dimension reduction control policy based on policy gradient learning

Yunfeng Gao[1], *Jianan Li*[1] *and Bingbing Pan*[1,*]

[1] Instrument and Electronics School of North University of China 030051, Taiyuan, China

SUMMARY: *In this paper, we propose a state space dimension reduction control policy optimization method based on policy gradient learning for continuous control modeling tasks in robot autonomous navigation in complex environments. In this method, the position, speed, course Angle, obstacle distance and target association information in the original navigation state are compressed and encoded to construct a low-dimensional state representation, which is jointly trained with the policy network to weaken the interference of redundant features on action search and enhance the stability and convergence efficiency of continuous control output. The experimental results on the simulation environment and the real robot platform show that compared with PPO, DDPG and the method without dimension reduction module, the navigation success rate of the proposed method reaches 96.7%, the average path length is reduced to 18.6 m, the decision delay is controlled at 0.041 s, and the training reward tends to be stable after 420 rounds. The real platform successfully reached the task target point in 19 out of 20 rounds of testing. Ablation experiments further show that the state space dimension reduction module has a significant support effect on the control smoothness, the control smoothness, the performance of complex scenes and the stability of dynamic obstacle avoidance, which can provide more stable strategy search boundaries and more efficient online deployment capabilities for robot navigation tasks.*

Povzetek: Ta članek predlaga metodo optimizacije strategije nadzora z redukcijo dimenzionalnosti prostora stanj za robotsko navigacijo, ki temelji na učenju z gradientom politike. Eksperimenti kažejo, da stopnja uspešnosti navigacije te metode doseže 96,7 %, povprečna dolžina poti se zmanjša na 18,6 m, zakasnitev odločanja je omejena na 0,041 s, stabilna konvergenca pa je dosežena po 420 iteracijah.

KEYWORDS: *Policy gradient learning; Robot navigation; State space dimension reduction; Control strategy optimization*

1 Introduction

With the expansion of mobile robots in scenarios such as warehousing and transportation, indoor services, and inspection collaboration, autonomous navigation systems have shifted from path generation units to computing frameworks with tightly coupled perception, decision making and control. In the process of navigation, the robot needs to process state information such as position and orientation, speed, heading, local obstacle distribution, target relative relationship and historical action feedback at the same time, and the dimension of state space continues to increase. Park et al. [1] verified the applicability of deep deterministic policy gradient methods for autonomous driving tasks of mobile robots in a sparse reward environment,

*18435131952@163.com
https://doi.org/10.65102/is2026095 Published: 30 April, 2026

and Lee and Yusuf[2] further illustrated the modeling value of deep reinforcement learning for mobile robot navigation. The state representation is too lengthy, and the policy network is prone to changes such as gradient update fluctuation, action search range expansion and sample utilization efficiency decline in the training phase, which affects the path smoothness and control consistency. Han et al. [3] analyzed the state modeling effect in collision avoidance from the perspective of self-state attention and sensing fusion, and Li et al. [4] pointed out that reward design and knowledge transfer would directly affect the stable formation of navigation strategy. In dynamic scenes and narrow channels, state redundancy also amplifies the disturbance of the decision boundary by invalid features, making it difficult for the navigation model to stably output a continuous control quantity.

Reinforcement learning provides a modeling path for autonomous navigation of mobile robots. Compared with traditional methods that rely on explicit environment models and manual rule design, policy gradient-based algorithms are able to directly learn continuous action mappings through environment interaction and form control policies under complex constraints. Wang et al. [5] demonstrated the adaptation ability of deep reinforcement learning to environmental interactive decision-making in landmark generation assisted navigation, and Chen and Liang[6] illustrated the application potential of continuous control framework in large-scale dynamic environments by improving DDPG path planning algorithm. In recent years, deep deterministic policy gradients, soft actor critics, and their improved frameworks have made progress in navigation success rate, path quality, and obstacle avoidance stability. Han et al. [7] improved the robot path planning method based on deep reinforcement learning, and Zhang and Chen[8] combined SAC and LSTM for dynamic indoor environment navigation, which all show that the strategy update mechanism has a strong support role for complex navigation tasks. Researches on graph relation reinforcement learning, multi-sensor fusion and risk map modeling show that the improvement of navigation performance depends on both policy update and state organization. Liu et al. [9] constructed a graph relationship reinforcement learning navigation framework in a large-scale crowded environment, and Zhao et al. [10] used an improved soft actor critic algorithm to improve the performance of mobile robot path planning.

When the original perception state is input into the policy network without screening, the model often needs to consume more rounds to distinguish the key features from the background noise, which is more obvious in the sparse reward and dynamic obstacle conditions. Tan[11] discussed the state input organization method in robot path planning from the perspective of multi-sensor information fusion. Xiao et al. [12] showed in the study of multi-modal fusion autonomous navigation that the quality of state expression under the condition of sparse reward would directly affect the efficiency of policy learning. Yang et al. [13] introduced the risk map reinforcement learning mechanism in the navigation of crowded environments, and Ou et al. [14] improved the autonomous navigation performance of mobile robots through the deep reinforcement learning method of sensor fusion. He et al. [15] focused on the autonomous navigation task in mapless and unknown environment, and pointed out that under the condition of insufficient environmental prior and continuous state disturbance, the navigation model put forward higher requirements for the integrity of state expression, strategy adaptation ability and online decision stability. Around high-dimensional state modeling, the current research evolution is mainly reflected in three levels. First, the state representation is shifted from hand-stitching to encoding, where position, distance, azimuths, and local geometric relationships are mapped into the feature space to enhance input consistency. Secondly, the control algorithm is shifted from policy updating to collaborative promotion of representation learning and policy learning, and the state compression, feature selection and memory mechanism are used to bind

the search boundary to reduce the interference of redundant dimensions on gradient estimation. Third, the deployment method moves from offline verification to online execution, and model training not only focuses on arrival rate and path length, but also emphasizes inference delay, convergence speed and entity platform migrability.

Based on the requirements of robot navigation, this paper proposes a state space dimension reduction control policy optimization method based on policy gradient learning. The method preserves pose change, target association and obstacle constraints in the state construction stage, compreses redundant features in the low-dimensional mapping stage, and establishes a joint learning mechanism of dimensionality reduction representation and continuous control output in the policy update stage. At the same time, the low-dimensional state representation can more directly correspond to the discrimination boundary required for action selection, and improve the stability and reliability of continuous control output. The work of this paper includes three aspects: constructing a more compact representation structure for navigation decision; The policy gradient update method coupled with the low-dimensional representation was designed. The navigation success rate, path length and online response performance are evaluated by simulation platform and real robot test, which provides calculation basis for autonomous navigation in complex environment.

2 Related work

In this section, three kinds of computer research in robot navigation are reviewed, including state space organization, policy gradient control and state representation compression, focusing on the representative methods for continuous decision-making of mobile robots in recent years. Table 1 summarizes the relevant literature from four aspects: core methods, state space treatment methods, control output characteristics, and association with this paper.

Table 1: Comparison of methods with literature related to robot navigation.

Literature	Core Method	State Space Processing Method	Control Output Characteristics	Relevance to This Study
Hu et al. [16]	Noisy N-step Dueling Double DQN	Directly inputs environmental observations and combines them with experience replay	High efficiency in discrete decision-making	Reflects the implementation path of value-function-based methods in computational navigation
Wong et al. [17]	PPO-based multi-robot navigation framework	Organizes multi-agent observation states	Relatively stable policy updates	Indicates that policy gradient methods are suitable for navigation control
Cheng et al. [18]	PPO-based physical-world learning framework	Integrates goal information with obstacle states	Can directly output continuous control actions	Demonstrates the practical deployment capability of policy models
Deshpande et al. [19]	DDPG-DG path planning method	Introduces differential game exploration information	Strong continuous action representation capability	Provides a reference for the control strategy design in this study
Tao and Kim [20]	Deep reinforcement learning-based local planning	Focuses on local navigation states in dynamic scenarios	Good online updating capability	Reflects the coupling between local decision-making and environmental variation
Yin et al. [21]	Off-policy reinforcement learning with curriculum learning	Gradually organizes the state space	Relatively stable convergence process	Indicates that state organization affects training performance
Cui et al. [22]	DRL hyper-heuristic sequential decision-making	Emphasizes state selection and policy scheduling	High decision-making flexibility	Suggests that high-dimensional states require effective screening mechanisms
Gao et al. [23]	Hierarchical reinforcement learning for mapless navigation	Hierarchically organizes navigation states and memory information	Suitable for long-range decision-making	Indicates that state hierarchy is beneficial for complex navigation
Montero et al. [24]	DRL-based dynamic warning zone method	Reconstructs states according to local risk regions	Sensitive to short-range control	Indicates that local state reconstruction affects control performance
Zhao et al. [25]	Explainable task-relevant state representation learning	Extracts task-relevant low-redundancy representations	Enhances the clarity of feature boundaries	Most closely aligned with the state-space dimensionality reduction focus of this study

In the research of robot navigation computer, the earlier methods mostly rely on discrete action search and local path modification. Hu et al. [16] improved the efficiency of path selection in unknown environments through noisy n-step update and preferential experience replay, but such value function models still have the characteristics of limited action expression in the scene of continuous coupling of speed and rotation Angle. In contrast, Wong et al. [17] applied PPO to the design of multi-robot navigation system, and Cheng et al. [18] verified the transferability of PPO framework on the entity platform, indicating that the policy gradient method is more suitable for establishing a direct mapping between high-dimensional states and continuous control variables. Deshpande et al. [19] added a differential game exploration mechanism to DDPG, and Tao and Kim[20] applied deep reinforcement learning to local path planning in dynamic environments. Together, these works show that continuous control modeling has become an important development direction of computer methods for robot navigation.

As the sensing dimension continues to increase, the state redundancy begins to directly affect the training efficiency and control stability of the policy network. Yin et al. [21] used curriculum learning to improve the off-policy navigation process in unknown environments, Cui et al. [22] enhanced sequential decision-making ability through hyper-heuristic deep reinforcement learning, Gao et al. [23] used hierarchical reinforcement learning to deal with long-range decision-making in mapless navigation. Montero et al. [24] enhanced local risk perception by means of dynamic warning area and short-range target design. The above researches have promoted the scene adaptation ability of computer navigation models. However, most of them pay more attention to reward construction, search mechanism or local decision structure, and less attention is paid to the compressed expression of high-dimensional state space. Zhao et al. [25] proposed a task-oriented interpretable state representation learning method, which provided a clearer feature boundary for model-independent deep reinforcement learning, and also showed that the collaborative modeling of state representation learning and policy update had high value. Based on this, the state space dimension reduction is incorporated into the optimization process of robot navigation control policy, and the joint calculation of state compression and action decision is completed under the policy gradient learning framework, so as to form a computer method that is more suitable for the continuous control requirements of complex scenes.

3 Method Introduction

3.1 Robot navigation task scene and state space composition

Mobile robot navigation usually includes two continuous steps: target reaching and local obstacle avoidance. In the warehouse corridor, service space and semi-structured experimental environment, the robot needs to reach the specified position according to the task sequence, and adjust the speed and heading in real time according to the change of the environment during the movement. This task requires the navigation system to continuously cover the target point and maintain a coherent path, while maintaining a stable control output in narrow areas and dynamic disturbance conditions. For the computer navigation model, the task scene is not only the path running space, but also the basic carrier of state sampling, feature organization and policy update.

In order to clearly illustrate the space composition, state source and computer representation relationship in the robot navigation task, Figure 1 shows the diagram of the robot navigation task scene and state space composition. In the figure, the geometric center of the robot is used as the reference point, and the target position, the obstacle boundary, the motion direction, and

the local passable area together form the spatial basis of the navigation state. The graph is not only used to describe the running position relationship of the robot in the environment, but also provides a unified computer modeling coordinate framework for subsequent state encoding, dimension reduction mapping and policy gradient control.

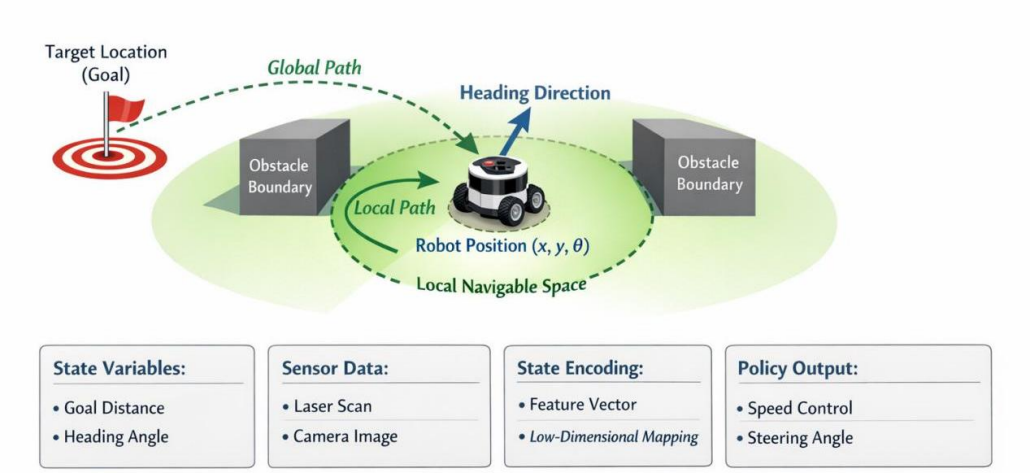

Figure 1: Schematic diagram of robot navigation task scene and state space composition.

Before establishing the whole navigation trajectory, the computer system needs to clarify the environment boundary, the target point position, the static obstacle distribution and the dynamic object activity range, and set the key state sampling item. Taking a rectangular indoor navigation space as an example, the geometric center coordinates of the robot are used as the reference origin, the forward direction is defined as the forward X-axis, the lateral offset is defined as the Y-axis, and the attitude Angle change corresponds to the heading state. The local obstacle distance, relative target orientation, linear velocity, angular velocity and historical action feedback together constitute the navigation state. Table 2 presents the main state items in the computer navigation model and their roles.

Table 2: Key components of the robot navigation state space.

No.	State Item	Specific Meaning
1	Current Position Coordinates	Represent the robot's real-time position in the global environment
2	Relative Distance to the Goal	Represent the distance relationship between the robot and the target point
3	Relative Direction to the Goal	Represent the offset angle of the target point relative to the current heading direction
4	Linear Velocity	Represent the robot's current forward speed
5	Angular Velocity	Represent the robot's current turning rate
6	Forward Obstacle Distance	Represent the nearest obstacle distance in the local space ahead of the robot
7	Lateral Obstacle Distance	Represent the local obstacle constraints on the left and right sides of the robot
8	Local Traversable Width	Represent the passable margin of the current corridor or local area
9	Historical Action Magnitude	Represent the influence of the control output at the previous time step on the current decision

At the computer implementation level, the state space is not a simple stack of sensing data, but is reorganized around navigation decision requirements. The position quantity describes the global task relationship, the obstacle quantity reflects the local traffic constraint, the motion quantity records control the execution trend, and the historical action term is used to supplement the time continuity. This composition helps the computer model distinguish valid states from background perturbations, and provides a more stable input boundary for subsequent policy gradient updates. When the robot enters the corner, narrow door and intersection area, the computer state vector can still maintain a uniform dimension, which is convenient for batch sampling, normalization processing and online real-time reasoning in the training phase, and reduces the state splicing error and the risk of drift.

The robot sequentially goes to the goal point and performs the navigation task in a planning sequence. During operation, the perception module composed of two-dimensional lidar, visual perception unit and odometer continuously collected the surrounding environment, and the computer module synchronously completed the status update, coordinate transformation and feature cache. The lidar is used to obtain the mid-range obstacle contour, the depth camera is responsible for supplemting the close-range geometric details, and the odometry provides the pose evolution information. Due to the high dimension of the navigation state, the original input often contains both task-related information and redundant disturbances. Therefore, the state space constructed in this section does not directly serve the traditional rule control, but serves as the unified input of the subsequent computer policy gradient network. Through the standardized organization of the state items, the navigation tasks can complete the environment representation, low-dimensional compression and continuous control mapping in the same computer framework, which provides a clear modeling basis for the state space dimension reduction and policy coupling mechanism in the next section.

3.2 Dimensionality reduction control method based on policy gradient learning

In this study, continuous control decisions in robot navigation scenarios are achieved by fusing the state-space dimensionality reduction mechanism with the policy gradient learning framework. When the robot runs in a complex environment, the computer system receives state information such as position, speed, heading Angle, target direction and local obstacle constraints. After dimensionality reduction coding, the compact features related to action selection are extracted, and the strategy is updated by combining the control results. As shown in Figure 2, the method link is composed of state input, dimension reduction mapping, policy sampling, action execution and reward update. The system obtains the reward value as the evaluation basis for navigation action adjustment, and obtains high reward in repeated interaction and iteration, so as to form a better control strategy.

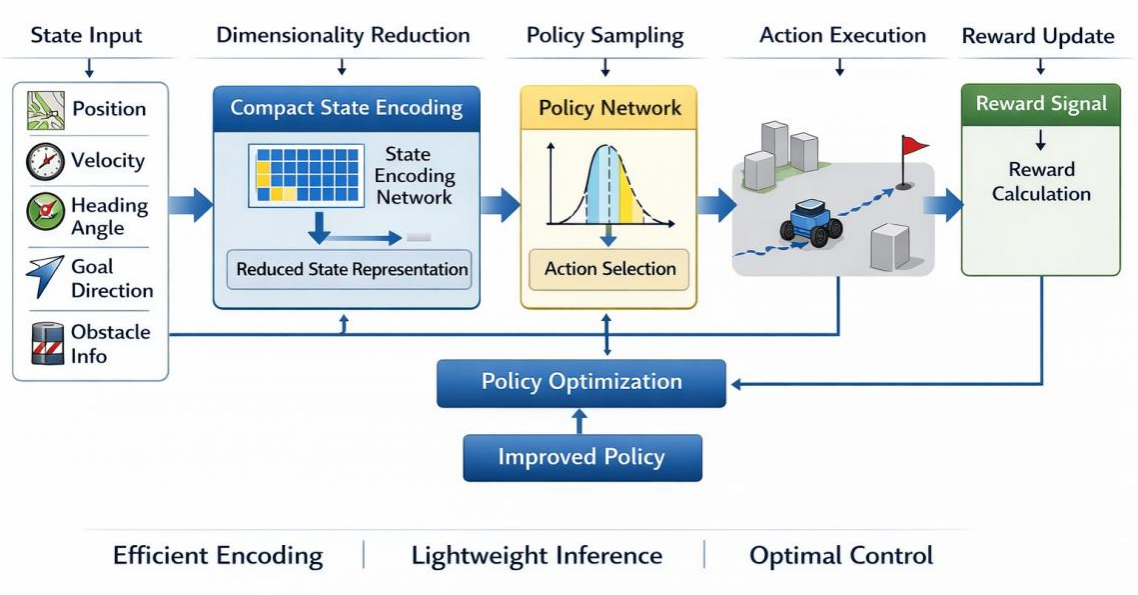

Figure 2: Schematic diagram of the dimensionality reduction control method based on policy gradient learning.

In addition, in view of the computing resource constraints of the navigation platform, in order to make the dimensionality reduction control method based on policy gradient learning more suitable for computer real-time reasoning, this paper introduces a lightweight design in the state encoding and policy update link. By constricting input dimensions, compressing redundant state components and reducing invalid feature propagation, the model parameter update process is more stable and the reasoning burden of the computer is controlled. After structural adjustment, the proposed method can improve the efficiency of action output while maintaining the decision-making accuracy, so as to complete stable path tracking and local obstacle avoidance control, and enhance the stability of model online deployment adaptation.

3.2.1 Original state space representation method

The basic principle of the original state space representation method is shown in Figure 3. In the process of robot navigation calculation, the computer needs to map the environment perception results, kinematic information and target constraints into a computable state vector. In order to maintain the consistency of the input structure of the computer, the current position, relative displacement of the target, linear velocity, angular velocity, local obstacle distance and historical control actions are organized into a set of original states. The state formed in this way not only retains the geometric relationship in the navigation task, but also provides a stable input boundary for subsequent computer dimensionality reduction coding and policy gradient update. Figure 3 shows the computer representation structure of the original state composed of global location relations, local passing constraints and action continuity information.

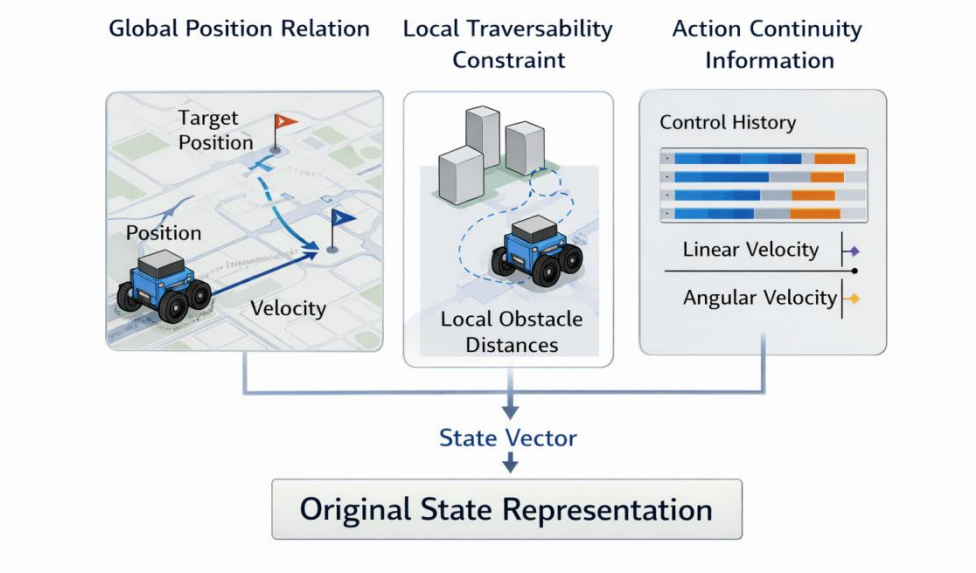

Figure 3: Schematic of the original state space representation method.

Suppose the position of the robot at time t is $p_t = (x_t, y_t)$, the target position is $g = (x_g, y_g)$, the linear and angular velocities are v_t and ω_t, respectively, and the forward, left and right obstacle distances are d_t^f, d_t^l, d_t^r, respectively. Then the original state vector can be expressed as follows:

$$s_t = [x_t, y_t, x_g - x_t, y_g - y_t, v_t, \omega_t, d_t^f, d_t^l, d_t^r, a_{t-1}] \tag{1}$$

In order to reduce the numerical interference of different dimensional states on the computer training process, the original states are normalized:

$$\hat{s}_t = \frac{s_t - \mu}{\sigma + \varepsilon} \tag{2}$$

Considering the temporal continuity of navigation control, the input representation of computer policy network is obtained by concatenating the current normalized state with the state at the previous time:

$$z_t = [\hat{s}_t, \hat{s}_{t-1}] \tag{3}$$

where x_t andy_t represent the current position of the robot in the computer map coordinate system; $x_g - x_t$ and $y_g - y_t$ represent the relative displacement of the target. v_t is the linear velocity; Let ω_t denote the angular velocity; d_t^f, d_t^l, d_t^r represent the local obstacle distances in the forward, left and right directions of the robot, respectively. a_{t-1} represents the control action at the previous time. μ and σ denote the mean and standard deviation of the training samples, respectively. A small constant for ε to prevent the denominator from becoming zero. z_t represents the final input vector fed into the computer policy network. Equation (1) is used to describe the basic composition of the original state space, Equation (2) is used to unify the numerical range of various state quantities, and Equation (3) is used to retain time continuous information in computer navigation control. After the above representation, the original state can simultaneously reflect the spatial relationship and control history among the robot, the target and the obstacle, so as to provide a directly computable input basis for the subsequent state space dimension reduction and policy gradient control.

3.2.2 Policy Gradient control algorithm

In policy-based reinforcement learning methods, the policy is usually approximated by a parameterized function, which has high adaptability in computer continuous control tasks. For robot navigation, the policy network needs to directly output the linear velocity increment and the angular velocity increment according to the current state. Therefore, this paper adopts the policy gradient control algorithm based on the Actor-Critic structure. Let the conditional distribution of the action at chosen by the computer policy in state zt be determined by the parameter θ. Then it is expressed as follows:

$$\pi_\theta(a_t|z_t) = p(a_t|z_t;\theta) \tag{4}$$

Here, $\pi_\theta(a_t|z_t)$ denotes the distribution of policies taking action a_t in state z_t; $p(a_t|z_t;\theta)$ is the conditional probability density controlled by the parameter θ. Let z_t denote the input state vector at time t; a_t denotes the control action at time t; Let θ denote the trainable parameters of the Actor network. Equation (4) is used to give the policy representation form in the policy gradient method, which provides a mathematical basis for subsequent action sampling and parameter updating.

In the continuous navigation scenario, the Actor network no longer enumerates discrete actions, but directly outputs the control action mean. The control quantity of the robot at time t can be expressed as follows:

$$a_t = \mu_\theta(z_t) + \xi_t \tag{5}$$

Here, $\mu_\theta(z_t)$ represents the deterministic control action output by the Actor network in state zt. Let ξ_t denote the exploration perturbation added in the training phase. a_t denotes the amount of continuous control that is finally executed. Equation (5) is used to describe the action generation mode of the robot in the computer training phase, so that the policy network not only maintains the continuous control ability, but also has the necessary exploration characteristics. The Actor network updates the parameters according to the value gradient returned by the Critic network, so as to improve the expected reward of the current policy.

During training, the robot gets an immediate reward r_t after performing an action and enters the next instant state z_{t+1}. In order to enhance the stability of computer training, this paper uses target value construction to update the target:

$$y_t = r_t + \gamma Q_{\phi^-}(z_{t+1}, \mu_{\theta^-}(z_{t+1})) \tag{6}$$

where y_t represents the target value at time t; r_t stands for immediate reward. Let γ denote the discount factor; $Q_{\phi^-}(\cdot)$ denotes the target Critic network; Let ϕ^- denote the network parameters of the target Critic; $\mu_{\theta^-}(z_{t+1})$ represents the action output by the target Actor network at the next time statez_{t+1}. Letθ^- denote the target Actor network parameters. Equation (6) is used to construct the time difference target value, so that the value estimation takes into account both current returns and future returns.

The Critic network completes the update by minimizing the mean square error, and its loss function is as follows:

$$L(\phi) = E[(Q_\phi(z_t, a_t) - y_t)^2] \tag{7}$$

Here, $L(\phi)$ represents the loss function of the Critic network; $E[\cdot]$ is the expected operation on the empirical sample. $Q_\phi(z_t, a_t)$ is the value estimate of the current Critic; y_t

represents the target value. Equation (7) is used to constrain the current value function to approach the target value, so as to improve the accuracy and training stability of computer value evaluation.

The update objective of the Actor network is to improve the expected reward of the current policy under the computer value evaluation, and its gradient form is as follows:

$$\nabla_\theta J(\theta) = E[\nabla_a Q_\phi(z_t, a)|_{a=\mu_\theta(z_t)} \nabla_\theta \mu_\theta(z_t)] \tag{8}$$

Here, $\nabla_\theta J(\theta)$ denotes the gradient of the policy objective function with respect to the parameter θ.$J(\theta)$ represents the policy optimization objective; $\nabla_a Q_\phi(z_t, a)|_{a=\mu_\theta(z_t)}$ said value function gradient of movement, and the action to take for $\mu_\theta(z_t)$ is calculated; Let $\nabla_\theta \mu_\theta \square z_t)$ denote the gradient of the Actor output with respect to the parameter θ. Equation (8) is used to pass the value evaluation result of Critic to Actor, so as to complete the update of policy parameters.

As shown in Figure 4, the computer implementation framework of the policy gradient control algorithm takes state input, policy sampling, action execution, reward return and parameter update as the main line to form a unified computer training link.

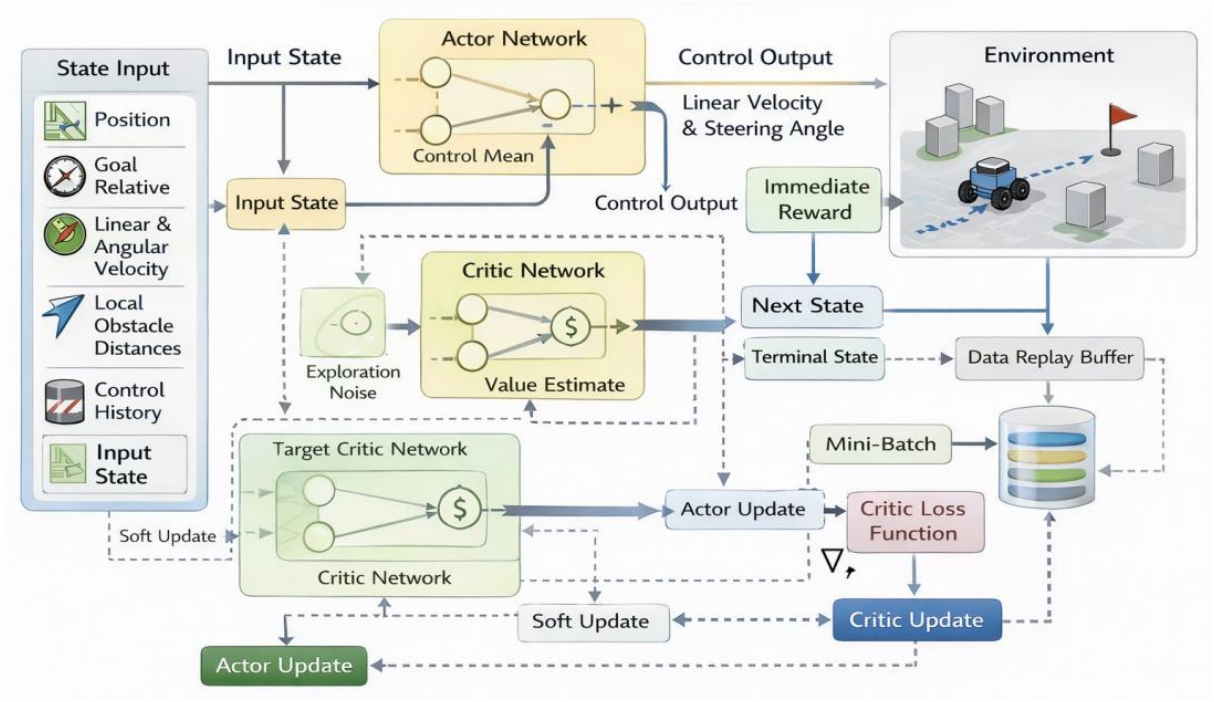

Figure 4: Policy gradient control algorithm implementation framework.

In addition, to avoid parameter oscillations, the target network is slowly synchronized with the online network using soft update. This mechanism can make the computer policy maintain a smoother convergence trajectory during the training phase, and weaken the influence of high-dimensional state perturbation on the control boundary. In the robot navigation task, the state space includes the current position, the relative direction of the target, the linear velocity, the angular velocity, the local obstacle distance and the historical action, and the action space is defined as the set of continuous velocity increment and angular increment. The reward function is composed of target approach reward, collision penalty and smooth control reward, which is used to guide the computer policy to form a balance between reaching efficiency and control stability. Compared with the traditional computer methods that only rely on local path

modification, the policy gradient control algorithm can directly establish a differentiable mapping between the state representation and the control output, so that the subsequent state space dimension reduction module not only serves to compress the feature, but also serves to narrow the policy search boundary and improve the efficiency of online deployment, which provides a unified computer decision-making basis for robot navigation control.

3.2.3 State space dimension Reduction and policy coupling mechanism

The state space dimension reduction module plays a connection role in environment representation and policy update, so that the robot can form a more stable continuous control output in complex navigation scenarios. Facing the original state input of the computer navigation model, the module compressed the position, velocity, target direction, local obstacle constraints and historical action values into a compact computer state representation through projection transformation, gated screening and residual retention. Compared with the direct input of high-dimensional states, this mechanism can weaken the interference of redundant features on the policy search boundary and enhance the correspondence between the state boundary and the action boundary. The main calculation process is as follows:

$$h_t = \mathrm{ReLU}(W_r e_t + b_r) \tag{9}$$

where e_t denotes the original state input vector at time t; W_r represents the dimension reduction mapping matrix; b_r denotes the bias term; h_t represents the intermediate feature after linear projection and nonlinear activation. $\mathrm{ReLU}(\cdot)$ denotes the linear rectified activation function. Equation (9) is used to map the high-dimensional original state into a more compact computational space while preserving the main discriminant features.

After completing the preliminary state projection, it is necessary to further judge the retention degree of each dimension feature in the current decision:

$$g_t = \sigma\left(W_g[e_t; a_{t-1}] + b_g\right) \tag{10}$$

where, W_g represents the gating mapping matrix; b_g represents the gating bias term; $[e_t; a_{t-1}]$ represents the concatenation vector of the current state and the previous action. a_{t-1} represents the control action at the previous time. g_t is the feature retention coefficient; Let $\sigma(\cdot)$ denote the Sigmoid activation function. Equation (10) is used to jointly determine the retention strength of each dimension feature according to the current state and historical control, so that the computer model has stronger context screening ability.

Based on the gated screening results, the model continues to fuse the residual information to maintain the structural integrity and distribution stability of the low-dimensional representation:

$$\hat{z}_t = \mathrm{LayerNorm}(g_t \odot h_t + \alpha e_t') \tag{11}$$

Here, $\hat{z}_t$ represents the low-dimensional state representation that is finally fed into the policy network. $\odot$ for element-wise multiplication; α represents the residual adjustment coefficient; e_t' represents the compressed residual term after matching the h_t dimension. ; $\mathrm{LayerNorm}(\cdot)$ represents the layer normalization operation. Equation (11) is used to retain part of the original structural information after gated screening and stabilize the low-dimensional state distribution, thus enhancing the representation consistency during computer training.

After the stable low-dimensional state representation is obtained, the policy network generates continuous control action output based on it:

$$a_t = \mu_\theta(\hat{z}_t) + \beta \tanh(W_a \hat{z}_t) \quad (12)$$

Here, $\mu_\theta(\hat{z}_t)$ represents the master control output of the Actor network in the low-dimensional state. W_a represents the action modification matrix; β represents the action adjustment coefficient; $\tanh(\cdot)$ denotes the hyperbolic tangent function; a_t denotes the final continuous control action. Equation (12) is used to superimpose the restricted correction term on the output of the basic strategy, so that the computer control quantity can maintain a smoother change trend in the speed and Angle update.

In order to ensure the collaborative convergence of state compression and action decision, a joint loss function should be constructed to constrain the training process:

$$L_c = -Q_\phi(\hat{z}_t, a_t) + \lambda_1 \|\hat{z}_t\|_1 + \lambda_2 \|a_t - a_{t-1}\|_2^2 \quad (13)$$

where, L_c represents the joint optimization objective of state dimension reduction and policy coupling phase. $Q_\phi(\hat{z}_t, a_t)$ is the value estimate of Critic network for low-dimensional state and action combination. Let λ_1 denote the sparsity constraint coefficient; $\|\hat{z}_t\|_1$ denotes the L_1 norm of the low-dimensional state vector; λ_2 denotes the action smoothness constraint coefficient; $\|a_t - a_{t-1}\|_2^2$ denotes the two-norm square between the current action and the previous action. Equation (13) is used to simultaneously constrain the value gain, state compactness and action continuity in computer training, so that the dimension reduction result and the control strategy form a more stable joint optimization relationship.

The above coupling mechanism enables the computer model to more clearly distinguish three types of control requirements during robot navigation: target approach, obstacle avoidance and steering adjustment. The weakly correlated quantities in the original state are compressed after gated screening, and the retained quantities directly participate in the policy update, making the action output of the Actor network smoother and the value estimate of the Critic network more stable. Since the input scale of low-dimensional state representation is smaller, the batch updates in computer training are more likely to form a consistent gradient, and the cache and computational overhead in online inference are also controlled. Therefore, state space dimension reduction is no longer just a front-end preprocessing step, but forms a unified computer decision link with policy gradient learning, which provides a clearer structural basis for subsequent convergence analysis, ablation experiments and real platform testing.

4 Experiment and analysis

4.1 Navigation environment Construction and model training

In this study, the mobile robot navigation environment is constructed in the Webots computer simulation platform, and the joint training of the state space dimension reduction module and the policy gradient control model is completed under the same computer framework. The scene is generated by the grid map editor and contains straight channel, corner area, narrow gate area, loop path and dynamic interference area. The robot uses a differential chassis, and the computer-aware input consists of laser ranging, odometry, target relative bearing, and historical action cache. To ensure the comparability between various computer models, the training scene boundary, sampling frequency, and reward scale are kept consistent. The main components of the navigation environment are shown in Table 3.

Table 3: Navigation environment build Settings.

Item	Value
Simulation Platform	Webots R2023b
Map Size	20 m × 20 m
Static Obstacle Units	18 groups
Dynamic Disturbance Units	6 groups
Number of Target Points	12
Control Period	0.1 s
State Update Frequency	10 Hz
Sensor Configuration	LiDAR + Odometry + Orientation Encoding
Action Form	Linear Velocity Increment and Angular Velocity Increment

In the model training phase, the robot performs environment exploration by the computer scheduling module, and the interaction samples are written into the experience pool according to state, action, reward and next state. The dimension reduction module is updated synchronally with the Actor and Critic networks, and the computer system records the average reward, arrival rate, collision rate, and action jitter amplitude every ten rounds for monitoring the convergence trend. In order to reduce the influence of random disturbance, the computer random seed is fixed and the input state is normalized during the training process. The core parameters for model training are shown in Table 4.

Table 4: Model training parameter Settings.

Parameter	Value
Actor Learning Rate	0.0001
Critic Learning Rate	0.001
Batch Size	256
Replay Buffer Capacity	100000
Discount Factor	0.99
Soft Update Coefficient	0.005
Maximum Training Episodes	1200
Steps per Episode	300
Validation Frequency	Once every 20 episodes

When the navigation success rate and average reward on the validation set remain stable, the computer system freezes the network parameters and moves to the independent testing phase. The test set consists of map layouts, obstacle combinations and target sequences that are not involved in the training, and the dynamic interference units contain different speeds and steering patterns. In the subsequent experiments, the basic PPO, DDPG and the method without dimension reduction module were used as benchmarks, and the comparative analysis was carried out from four dimensions of navigation success rate, path length, decision delay and control smoothness. The training server is configured with RTX4080 graphics card and 32GB memory to ensure that the computer training and reasoning process has stable efficiency support.

4.2 Simulation results and comparative experiments

The dimension reduction control method of robot navigation state space based on policy gradient learning shows strong path generation ability and control stability in computer simulation. The variation of the navigation success rate of different models in successive rounds was recorded in the training phase. The computer statistical results show that after adding the

state space dimension reduction, the model still has some fluctuations in the early stage, but the recovery speed is faster in the middle and late stage, and the rise of success rate is more concentrated, which indicates that low-dimensional state representation is helpful for the bandaging strategy to search the boundary and enhance the consistency of gradient update. Compared with the basic PPO and DDPG, the proposed method enters the stable interval earlier and the training trajectory is smoother under complex obstacle layouts.

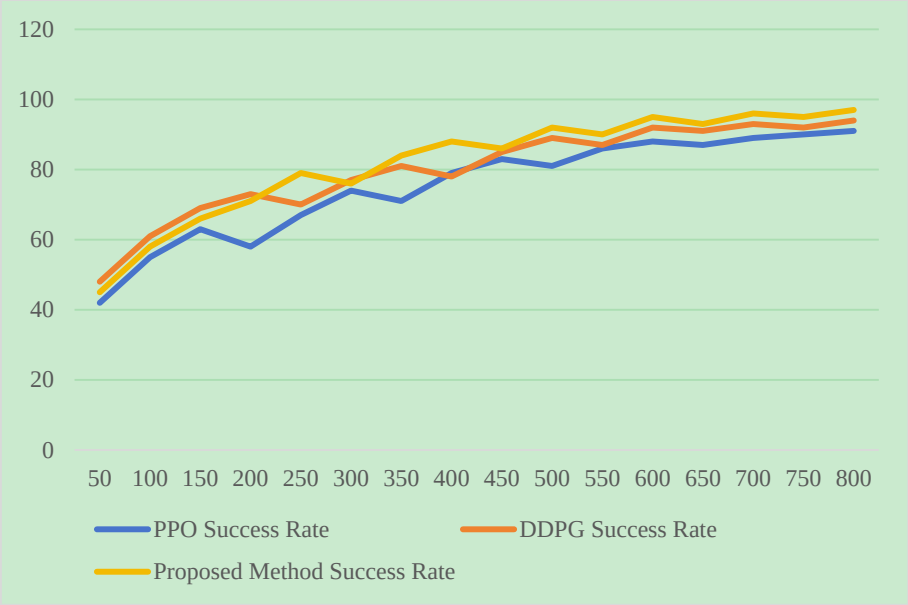

Figure 5: Variation of navigation success rate under training rounds.

As shown in Figure 5, the basic PPO has obvious ups and downs in the middle and front segments. Although DDPG is higher than PPO as a whole, it also falls back in local rounds. The proposed method gradually opened the gap after 350 rounds and maintained a high level in subsequent training. The computer playback results show that the dimension reduction module strengthens the role of the relative direction of the target, the forward obstacle distance and the historical action quantity in the state representation, and reduces the invalid trial in the continuous action update of the policy network, thereby improving the success rate and shortening the fluctuation interval in the convergence stage.

The test phase further compares the average path cost of different models in independent scenarios. The computer system synthesizes the collision risk, path length, and action adjustment amplitude in each scene into a unified cost value, which is used to measure the overall navigation quality of the model in a new scene. The results show that the proposed method can maintain a low cost in most test scenarios, especially in the corner area, narrow gate area and dynamic interference cut scene, the control boundary is more stable, and the number of detour is significantly reduced.

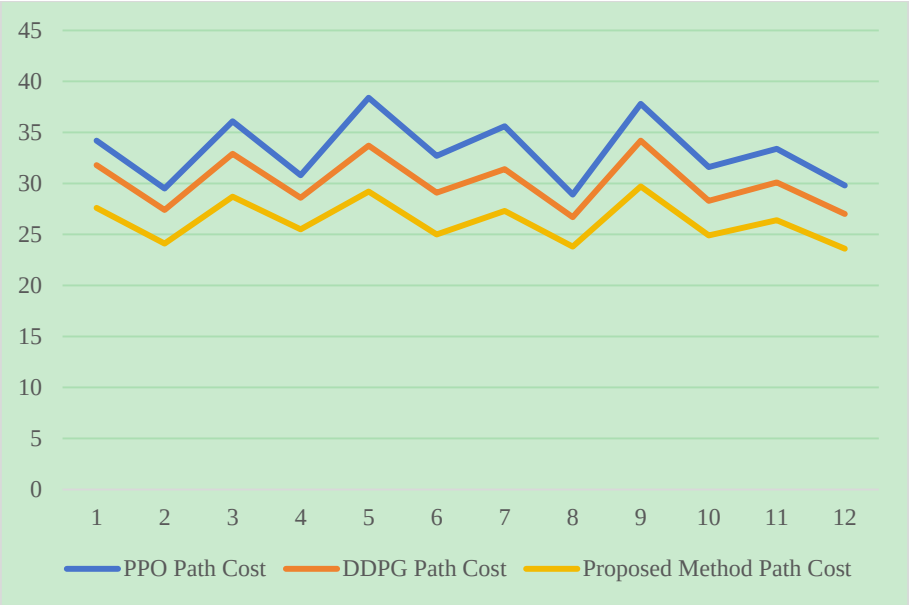

Figure 6: Average path cost variation in the test scenario.

As shown in Figure 6, the cost of PPO fluctuates greatly in multiple scenarios, and although DDPG decreases, it still increases in high complexity scenarios. The overall curve of the proposed method is always at a low position, and the fluctuation range is smaller, indicating that the state space dimension reduction not only reduces the redundant interference in the computer reasoning link, but also improves the quality of value transfer between the Actor and the Critic, so that the control action maintains higher consistency in the continuous update.

To quantitatively compare the overall performance of the three methods, Table 5 presents the main metrics on independent test sets. The results show that the navigation success rate of the proposed method reaches 96.7%, which is 7.9 percentage points higher than that of basic PPO and 4.8 percentage points higher than that of DDPG. The average path length is reduced to 18.6 m, the average path cost is 24.9, the decision delay is controlled at 0.041 s, and the control jitter amplitude is also kept at a low level. Statistical tests show that the proposed method and the two types of baselines achieve significant difference levels in the main indicators.

Table 5: Comparison of navigation indicators of different models on the test set (mean ± standard deviation).

Metric	PPO	DDPG	Proposed Method
Navigation Success Rate (%)	88.8 ± 3.4	91.9 ± 2.8	96.7 ± 1.9
Average Path Length (m)	22.7 ± 1.8	20.9 ± 1.5	18.6 ± 1.1
Average Path Cost	31.4 ± 2.3	28.8 ± 2.0	24.9 ± 1.4
Decision Latency (s)	0.063 ± 0.008	0.052 ± 0.006	0.041 ± 0.004
Control Jitter Amplitude	0.38 ± 0.05	0.31 ± 0.04	0.22 ± 0.03
Significance Level (p)	—	< 0.05	< 0.01

In general, the method in this paper forms a more balanced computer control performance between security, economy and real-time. In the open area, the three methods can complete the target arrival, but the path of the proposed method is less curved and the cumulative Angle is lower. In the area with dense obstacles, the basic PPO is prone to have too many local trials, and the DDPG has short-time delay when the dynamic interference enters. However, the

proposed method can lock the direction faster with the help of low-dimensional state representation. A small number of failure samples are concentrated in the combination scenario where the two dynamic bodies cross close to each other and the channel width is rapidly reduced, so the cost will briefly increase, but the model can still recover stability in the subsequent step, which shows that the computer framework has good scene adaptation ability.

4.3 Convergence performance and ablation experiments

In order to evaluate the convergence characteristics and structural contribution of the proposed method in the computer training phase, the average reward, navigation success rate, control jitter amplitude and decision delay are continuously recorded in the computer environment, and the removal test of key modules is carried out. The results show that the full model enters the stable interval after 420 rounds, the average return grows faster than each simplified structure, and the subsequent fluctuation range is smaller, which indicates that the state space dimension reduction and policy coupling mechanism can compress the invalid search process and enhance the consistency of gradient transfer between Actor and Critic. The computer training log also shows that the success rate of the full model is improved more continuously in the middle and late segments, and the control output changes more smoothly, indicating that the framework forms a better coordination relationship between the convergence speed and the control quality.

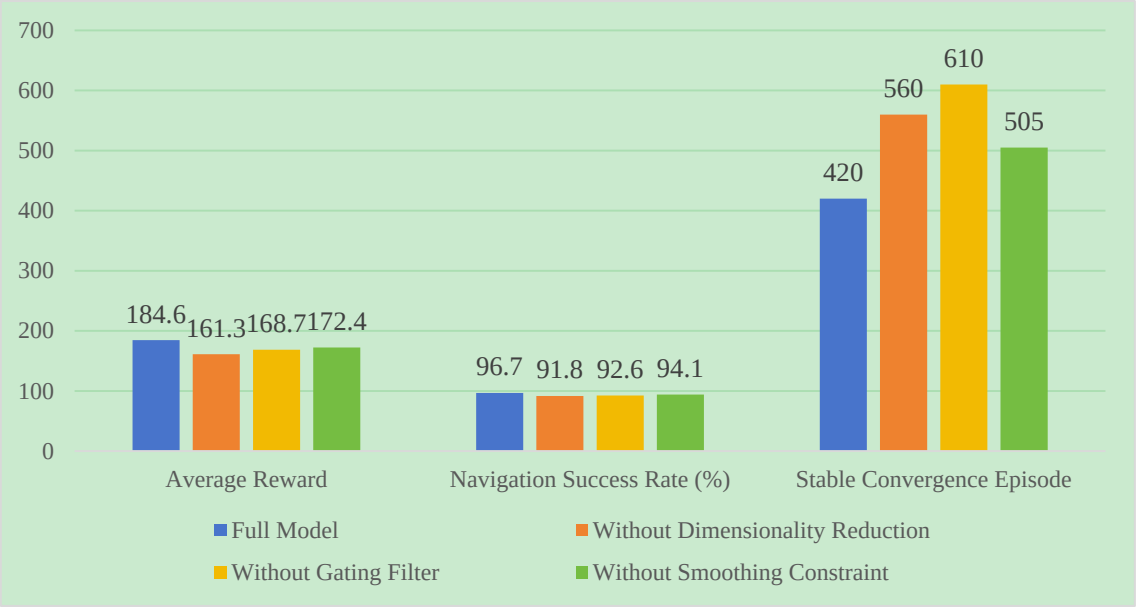

Figure 7: Comparison of model convergence performance.

As shown in Figure 7, the full model outperforms the de-dimensionality reduction module, the de-gated screening and the de-smoothing constraint structure in the three dimensions of navigation success rate, average return and stable convergence rounds. The average return of the full model reaches 184.6, which is significantly higher than that of the other three types of structures. The navigation success rate reached 96.7%, which was 4.9 percentage points higher than that of the de-dimensionality reduction module and 4.1 percentage points higher than that of the de-gating filter structure. At the same time, the complete model enters the stable convergence interval at the 420th round, which is earlier than the other three types of structures, indicating that the state compression and gated screening in the process of computer training can effectively reduce the disturbance caused by high-dimensional redundant inputs. After the dimensionality reduction module was removed, the success rate and reward decreased

simultaneously, and the stable convergence rounds were delayed to 560 rounds, indicating that the original high-dimensional state would expand the strategy search boundary. After de-gating screening, the average return remains in a high range, but the stable convergence rounds are postponed to 610 rounds, which indicates that the weight allocation of key features has a direct impact on the convergence efficiency of the computer. After removing the smoothing constraint, the decrease in success rate is relatively small, but the stable convergence rounds are still later than that of the full model, indicating that the action continuity constraint has an obvious support effect on the stable output in the later stage of training.

To further quantify the performance differences of different structures, Table 6 presents the results of ablation experiments. After removing the dimensionality reduction module, the navigation success rate decreases by 4.9%, the average path cost increases by 11.6%, and the computer decision delay increases from 0.041 s to 0.049 s, which indicates that high-dimensional redundant states will weaken the efficiency of policy update. After removing the gated screening, the control jitter amplitude expands to 0.31, indicating that the computer model is more unstable in its response to local obstacle changes. After removing the action smoothness constraint, although the arrival rate only decreases slightly, the cumulative turn Angle increases significantly and the path continuity becomes weaker. The complete model remains optimal in the success rate, cost and time delay, which indicates that the computer framework forms a more robust balance between convergence speed and control quality.

Table 6: Comparison of the results of ablation experiments.

Model	Success Rate (%)	Path Cost	Latency (s)	Jitter Amplitude
Full Model	96.7	24.9	0.041	0.22
Without Dimensionality Reduction	91.8	27.8	0.049	0.27
Without Gating Filter	92.6	26.9	0.046	0.31
Without Smoothing Constraint	94.1	26.1	0.044	0.29

To sum up, the state space dimension reduction is responsible for compressing computer input redundancy, the gated screening is responsible for stabilizing the distribution of key features, and the smoothing constraint is responsible for controlling the continuous change of actions. The three jointly support the convergence and deployment adaptability of the computer navigation model. A small number of failure samples mainly appear in the scenario where the double dynamic body is crossing approximation and the channel width is rapidly shrinking, in which case the return will briefly decrease, but the full model can still recover stability in the subsequent step.

4.4 Real robot platform test

In order to verify the executability of the proposed method in a real robot platform, this paper carries out field tests in a composite scene composed of an indoor and outdoor transition corridor and an open experimental area. The total length of the test area is 52 m, including three narrow doors, two right-angle turns and four groups of dynamic interference units, and the average running speed of the robot is set to 0.8 m/s. The actual measurement platform uses a differential mobile robot chassis, equipped with an industrial camera, a two-dimensional lidar, a wheeled odometer and a Jetson Orin computer module. The computer side memory is 16 GB, and the control frequency is maintained at 20 Hz. In order to ensure the comparison value of the test results, the robot completed 20 rounds of navigation tasks continuously under the same starting and ending conditions, and recorded the path deviation, decision delay, local obstacle

avoidance action and task completion time. The running results of the robot in the real platform are shown in Figure 8.

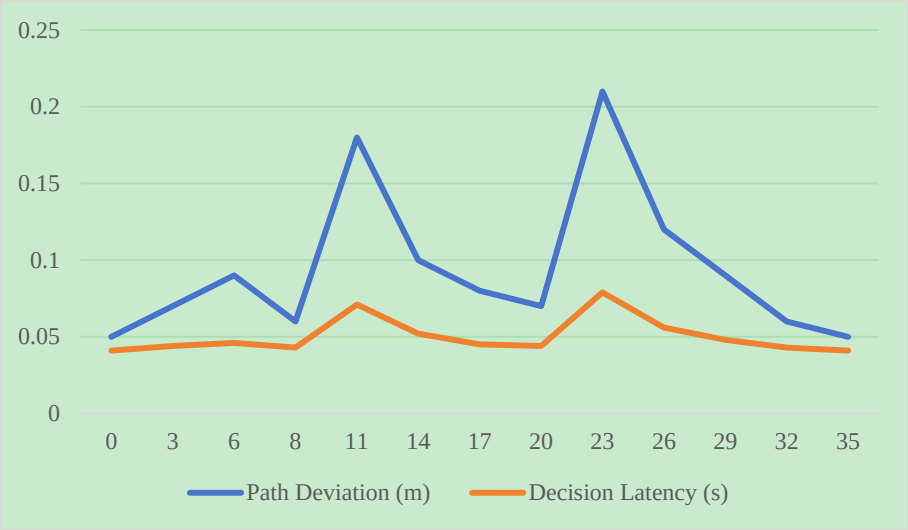

Figure 8: Path deviation versus decision delay variation in real robot platform test.

The figure shows the changes of path tracking error and online decision delay in the process of single round of testing. It can be seen that the computer control link will have short-term fluctuations when the robot enters the narrow gate area and the corner area, but after the participation of the state space dimension reduction module, the duration of error growth is shorter and the time delay decreases faster. The path error remained within 0.10 m in most of the time, and only increased significantly twice around the 11th s and 23rd s of the dynamic disturbance, with the peak values reaching 0.18 m and 0.21 m respectively, and then returned to the stable range within three control periods. The computer log shows that the policy gradient network still maintains good action continuity in the field test, and there is no frequent reverse correction.

From the overall results, 19 of the 20 rounds of tasks successfully reached the target point, the average path length was 54.3 m, the average task completion time was 68.5 s, and the average decision delay of the computer was 0.047 s. Compared with the control structure without dimension reduction module, the average path deviation of the proposed method is reduced by 13.6%, the cumulative Angle is reduced by 11.2%, and the number of local pauses is reduced from 2.4 times per round to 1.1 times. The field playback results show that the low-dimensional state representation reduces the computer input redundancy, so that the control strategy can still maintain high consistency in the presence of real sensing noise. A small number of failure samples appear in the combination condition of double dynamic body crossing approach accompanied by ground reflection interference. At this time, the computer perception update lags slightly, but the robot can still complete safe deceleration and detour through subsequent strategy correction. The overall results show that the computer model has good stability in field migration and deployment.

5 Discussion

Field test and simulation results show that the proposed method maintains high consistency and stability in complex navigation scenes. In the independent test set, the navigation success rate reached 96.7%, which was 7.9 percentage points higher than that of PPO and 4.8 percentage

points higher than that of DDPG. The average path length is reduced to 18.6 m, the average path cost is 24.9, the decision delay is controlled at 0.041 s, and the control jitter amplitude is 0.22. In the real robot platform test, 19 rounds of 20 rounds of tasks successfully reached the target point, the average path length was 54.3 m, the average task completion time was 68.5 s, and the average decision delay of the computer was 0.047 s. Compared with the control structure without dimension reduction module, the average path deviation is reduced by 13.6%, the cumulative Angle is reduced by 11.2%, and the number of local pauses is reduced from 2.4 times per round to 1.1 times. The above results show that the state space dimensionality reduction does not weaken the environment discrimination ability, but improves the effective information density in the computer input. The performance gain mainly comes from two aspects. On the one hand, the dimensionality reduction module compresses the weak correlation in the high-dimensional state, so that the computer policy network can use the target direction, obstacle distance and historical action information more centrally, so as to shorten the search path and reduce the invalid correction. On the other hand, the dimension reduction representation and policy gradient update are co-modeled in a unified computer framework, and the gradient transfer between value assessment and action generation is smoother, so that the training convergence speed, online inference efficiency and control smoothness are improved at the same time. The adaptation ability and operation stability of the model in field tests are also better.

6 Conclusions

This paper proposes a strategy optimization method based on policy gradient learning to reduce the dimension of robot navigation state space, which realizes the integrated computer modeling of state compression, continuous control and online decision-making in complex scenes. In this method, the position, velocity, target direction, local obstacle constraint and historical action amount are compressed into a compact representation, and the collaborative update of value assessment and action generation is completed within a unified computer framework. The simulation results show that the navigation success rate reaches 96.7%, the average path length is reduced to 18.6 m, the decision delay is controlled at 0.041 s, and the control jitter amplitude is maintained at 0.22. In the real robot platform test, 19 rounds of 20 rounds of tasks successfully reached the target point, the average path deviation was reduced by 13.6%, the cumulative rotation Angle was reduced by 11.2%, and the number of local pauses was reduced from 2.4 times per round to 1.1 times, indicating that the state representation formed by the method in the computer training stage has good transfer ability and field adaptability. The coupled design of state space dimension reduction and policy gradient update makes the computer model form a relatively stable balance between convergence speed, inference efficiency and control smoothness, and also provides a reusable structure for computer decision-making in complex navigation tasks. Future work can continue to focus on multi-robot cooperative navigation, 3D dynamic scene modeling, and lightweight computer deployment, so as to enhance the operation stability and generalization ability of the model in largescale environments. Ablation results further show that the dimensionality reduction module, gated screening and smoothing constraints jointly support the stable training and real-time deployment of the computer navigation model, and also enhance the adaptation ability and operation stability of the model in engineering applications.

Funding

This work was supported by Graduate Education Innovation Program of Shanxi Province (2025SJ029).

About the Author

Yunfeng Gao received the B.S. degree in communication engineering from the North University of China, Taiyuan, China, in 2022, where he is currently working towards the M.S. Degree in instrument and meter engineering. His main research is learning-based control, disturbance rejection for hypersonic morphing vehicles.

Jianan Li received his Bachelor of Engineering degree from Lanzhou Jiaotong University in Lanzhou, China in 2022. Currently, he is pursuing a Master of Engineering degree in Instrument Science and Technology at North University of China in Taiyuan, China. His main research interests focus on electrical impedance tomography and defect location recognition and detection.

Bingbing Pan was born in Linfen, Shanxi, China, in 1996. She received the B.S. degree in electronic science and technology from North University of China, Taiyuan, China, in 2017, and the M.S. degree in optics from University of Science and Technology of China, Hefei, China, in 2020. She is currently pursuing the Ph.D. degree in School of Instrument and Electronics in North University of China. Her main research direction is advanced guidance algorithms for hypersonic gliding vehicles.

References

[1] Park M, Lee S Y, Hong J S, et al. Deep deterministic policy gradient-based autonomous driving for mobile robots in sparse reward environments[J]. Sensors, 2022, 22(24): 9574. https://doi.org/10.3390/s22249574

[2] Lee M F R, Yusuf S H. Mobile robot navigation using deep reinforcement learning[J]. Processes, 2022, 10(12): 2748.https://doi.org/10.3390/pr10122748

[3] Han Y, Zhan I H, Zhao W, et al. Deep reinforcement learning for robot collision avoidance with self-state-attention and sensor fusion[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 6886-6893.https://doi.org/10.1109/LRA.2022.3178791

[4] Li W, Yue M, Shangguan J, et al. Navigation of mobile robots based on deep reinforcement learning: Reward function optimization and knowledge transfer[J]. International Journal of Control, Automation and Systems, 2023, 21(2): 563-574. https://doi.org/10.1007/s12555-021-0642-7

[5] Wang X, Sun Y, Xie Y, et al. Deep reinforcement learning-aided autonomous navigation with landmark generators[J]. Frontiers in Neurorobotics, 2023, 17: 1200214. https://doi.org/10.3389/fnbot.2023.1200214

[6] Chen Y, Liang L. SLP-improved DDPG path-planning algorithm for mobile robot in large-scale dynamic environment[J]. Sensors, 2023, 23(7): 3521. https://doi.org/10.3390/s23073521

[7] Han H, Wang J, Kuang L, et al. Improved robot path planning method based on deep reinforcement learning[J]. Sensors, 2023, 23(12): 5622. https://doi.org/10.3390/s23125622

[8] Zhang Y, Chen P. Path planning of a mobile robot for a dynamic indoor environment based on an sac-lstm algorithm[J]. Sensors, 2023, 23(24): 9802. https://doi.org/10.3390/s23249802

[9] Liu Z, Zhai Y, Li J, et al. Graph relational reinforcement learning for mobile robot navigation in large-scale crowded environments[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(8): 8776-8787. https://doi.org/10.1109/TITS.2023.3269533

[10] Zhao T, Wang M, Zhao Q, et al. A path-planning method based on improved soft actor-critic algorithm for mobile robots[J]. Biomimetics, 2023, 8(6): 481. https://doi.org/10.3390/biomimetics8060481

[11] Tan J. A method to plan the path of a robot utilizing deep reinforcement learning and multi-sensory information fusion[J]. Applied Artificial Intelligence, 2023, 37(1): 2224996. https://doi.org/10.1080/08839514.2023.2224996

[12] Xiao W, Yuan L, Ran T, et al. Multimodal fusion for autonomous navigation via deep reinforcement learning with sparse rewards and hindsight experience replay[J]. Displays, 2023, 78: 102440. https://doi.org/10.1016/j.displa.2023.102440

[13] Yang H, Yao C, Liu C, et al. Rmrl: Robot navigation in crowd environments with risk map-based deep reinforcement learning[J]. IEEE Robotics and Automation Letters, 2023, 8(12): 7930-7937.https://doi.org/10.1109/LRA.2023.3322093

[14] Ou Y, Cai Y, Sun Y, et al. Autonomous navigation by mobile robot with sensor fusion based on deep reinforcement learning[J]. Sensors, 2024, 24(12): 3895. https://doi.org/10.3390/s24123895

[15] He N, Yang Z, Bu C, et al. Learning autonomous navigation in unmapped and unknown environments[J]. Sensors, 2024, 24(18): 5925.https://doi.org/10.3390/s24185925

[16] Hu W, Zhou Y, Ho H W. Mobile robot navigation based on noisy N-step dueling double deep Q-network and prioritized experience replay[J]. Electronics, 2024, 13(12): 2423. https://doi.org/10.3390/electronics13122423

[17] Wong C C, Weng K D, Yu B Y. Multi-Robot Navigation System Design Based on Proximal Policy Optimization Algorithm[J]. Information, 2024, 15(9): 518. https://doi.org/10.3390/info15090518

[18] Cheng W C, Ni Z, Zhong X, et al. Autonomous robot goal seeking and collision avoidance in the physical world: An automated learning and evaluation framework based on the ppo method[J]. Applied Sciences, 2024, 14(23): 11020. https://doi.org/10.3390/app142311020

[19] Deshpande S V, Harikrishnan R, Ibrahim B S K S M K, et al. Mobile robot path planning

using deep deterministic policy gradient with differential gaming (DDPG-DG) exploration[J]. Cognitive Robotics, 2024, 4: 156-173. https://doi.org/10.1016/j.cogr.2024.08.002

[20] Tao B, Kim J H. Deep reinforcement learning-based local path planning in dynamic environments for mobile robot[J]. Journal of King Saud University-Computer and Information Sciences, 2024, 36(10): 102254. https://doi.org/10.1016/j.jksuci.2024.102254

[21] Yin Y, Chen Z, Liu G, et al. Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning[J]. Expert Systems with Applications, 2024, 247: 123202. https://doi.org/10.1016/j.eswa.2024.123202

[22] Cui T, Yang X, Jia F, et al. Mobile robot sequential decision making using a deep reinforcement learning hyper-heuristic approach[J]. Expert Systems with Applications, 2024, 257: 124959.https://doi.org/10.1016/j.eswa.2024.124959

[23] Gao Y, Lin F, Cai B, et al. Mapless navigation via Hierarchical Reinforcement Learning with memory-decaying novelty[J]. Robotics and Autonomous Systems, 2024, 182: 104815.https://doi.org/10.1016/j.robot.2024.104815

[24] Montero E E, Mutahira H, Pico N, et al. Dynamic warning zone and a short-distance goal for autonomous robot navigation using deep reinforcement learning[J]. Complex & Intelligent Systems, 2024, 10(1): 1149-1166. https://doi.org/10.1007/s40747-023- 01216-y

[25] Zhao T, Li G, Zhao T, et al. Learning explainable task-relevant state representation for model-free deep reinforcement learning[J]. Neural Networks, 2024, 180: 106741. https://doi.org/10.1016/j.neunet.2024.106741