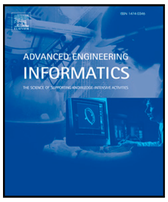
Advanced Engineering Informatics 72 (2026) 104379

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Full length article

Human–machine hybrid deep reinforcement learning for autonomous navigation in unknown environments[☆]

Yongheng Li[a], Qianqian Zhang[a,*], Yu Kang[b], Yun-Bo Zhao[c]

[a] *Anhui University, School of Artificial Intelligence, Hefei, 230601, Anhui, China*
[b] *Hefei University of Technology, School of Electrical Engineering and Automation, Hefei, 230009, Anhui, China*
[c] *University of Science and Technology of China, Department of Automation, Hefei, 230026, Anhui, China*

ARTICLE INFO

Keywords:
Human–machine hybrid control
Deep reinforcement learning (DRL)
Reactive navigation
Complex and unknown environments

ABSTRACT

Navigating complex, unknown environments poses a significant challenge for autonomous systems, where traditional deep reinforcement learning (DRL) models frequently stagnate in local optima. Meanwhile, existing human–machine hybrid approaches remain constrained by rigid interaction mechanisms and inefficient experience integration, lacking adaptable co-decision frameworks and effective human-guided local target point (LTP) decomposition. In response to these limitations, this paper proposes a human–machine hybrid deep reinforcement learning (HM-DRL) method for autonomous navigation. Specifically, a human–machine co-decision mechanism (H-MCDM) is introduced in the model training process, which rationally allocates control authority between humans and the machine, thereby replacing the rigid substitution-based human–machine interaction mode. Building on this mechanism, to enhance the optimization capability of reactive navigation in long-distance tasks, we propose a human-guided LTP selection and evaluation method. This approach filters the set of local target points (SLTP) based on real-time sensor data and selects the optimal LTP to decompose long-range navigation. In this framework, human analytical and predictive capabilities are leveraged to provide real-time dynamic guidance via intervention in the LTP selection process, realizing collaboration across both training and decision-making phases. Experimental results demonstrate that the proposed method outperforms traditional DRL methods, achieving improved navigation performance and enhanced global optimization capability.

1. Introduction

With the advancement of society, autonomous navigation technology is increasingly applied in complex and unknown environments, including autonomous driving, logistics, and warehouse management [1–3]. These tasks require robots to plan collision-free paths using environmental information. While global path planning methods, such as A*, D*, and PRM, are effective in static environments, their performance is degraded in dynamic or unknown settings [4–6]. Traditional reactive navigation methods, which do not rely on global maps, are constrained by poor robustness and challenges in parameter and rule tuning [7]. For instance, methods like the dynamic window approach (DWA) and artificial potential field (APF) depend on real-time local information and predefined rules, limiting their adaptability due to the inability to utilize historical experience or predict future states [8,9]. In contrast, deep reinforcement learning (DRL) learns complex navigation strategies through neural networks, allowing for adaptation to unknown environments and continuous optimization of navigation performance [10,11].

As a reactive navigation approach, DRL does not require prior knowledge or manual rule design, iteratively optimizing strategies through continuous interaction with the environment [12,13]. However, in complex and unknown environments, DRL-based navigation frequently stagnates in local optima during long-distance tasks, as the absence of global information impedes the evaluation of long-term decision impacts [14,15]. For example, agents often become trapped by U-shaped obstacles, failing to reach global goals [16]. While researchers have attempted to mitigate this, methods based on reward shaping or safety control strategies either suffer from limitations such as inadequate global path awareness, over-reliance on rules, poor adaptation to dynamic obstacles, compromised efficiency, or inability to decompose long-distance tasks [17–20]. In contrast, human drivers

[☆] This article is part of a Special issue entitled: ‘Human-AI Collaboration’ published in Advanced Engineering Informatics.
[*] Corresponding author.
E-mail addresses: WA23301086@stu.ahu.edu.cn (Y. Li), qqzhang@ahu.edu.cn (Q. Zhang), kangduyu@hfut.edu.cn (Y. Kang), ybzhao@ustc.edu.cn (Y.-B. Zhao).

https://doi.org/10.1016/j.aei.2026.104379
Received 26 March 2025; Received in revised form 27 November 2025; Accepted 21 January 2026
Available online 13 February 2026

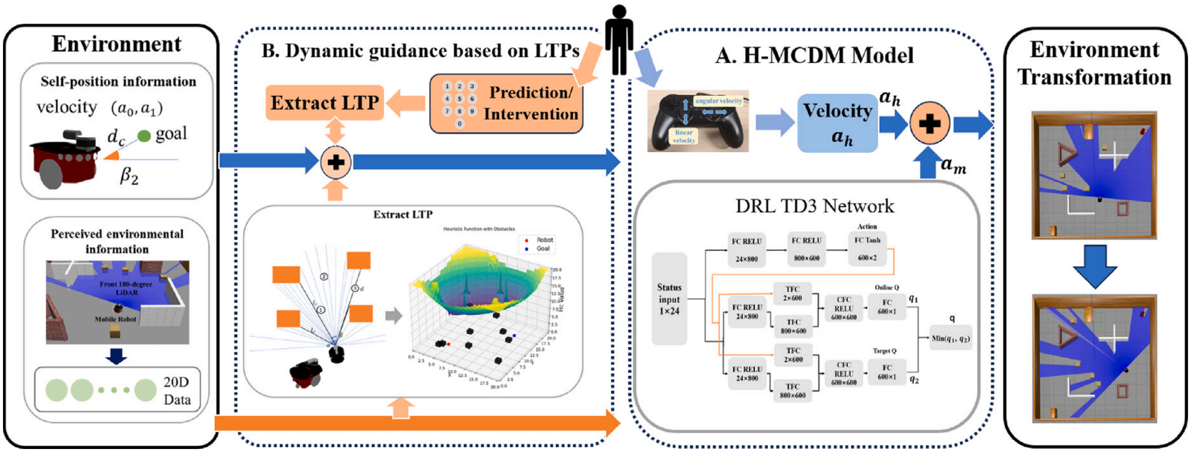


Fig. 1. The overall framework of the proposed HM-DRL method is illustrated below. (a) We introduce an H-MCDM to appropriately allocate control authority, integrating human expertise with the self-learning capabilities of DRL to train the TD3 model. (b) To address the local optima issue in reactive navigation for long-range tasks, we design a method for LTP selection and evaluation. In addition, human dynamic guidance is incorporated into the LTP selection process, embedding human predictive capabilities into the robot's decision-making model for unknown environments.

can leverage experience to predict future paths and reach destinations even with limited cues [21]. Human–machine hybrid intelligence has already demonstrated advantages in fields like healthcare, industrial manufacturing, and warehouse management [22–24]. Particularly in autonomous driving, predicting the behavior of surrounding vehicles and pedestrians is crucial. Although AI-based prediction has made considerable progress, humans still retain advantages in cue interpretation and intention inference in certain complex and dynamic scenarios [25, 26].

In existing human–machine hybrid deep reinforcement learning, human experience is mainly integrated through experience replay, experience buffer division, or the complete replacement of machine actions [27–30]. However, these approaches exhibit significant limitations. First, relying on partitioned expert buffers involves pre-collected static samples, where only the sampling timing and ratio can be adjusted; this prevents the flexible injection of real-time human experience during the training process, resulting in poor adaptability. Second, the strategy of completely substituting machine actions with human inputs is overly rigid, often necessitating frequent pauses to await human intervention, which leads to disruptive training interruptions. Beyond these action-level limitations, we note that structural optimization is also critical. In fields such as micro aerial vehicles (MAVs) and unmanned aerial vehicles (UAVs), task decomposition using local target points (LTPs) has been proven effective in alleviating the local optima problem in autonomous navigation [31–34]. Despite these advancements, the generation of such targets often relies on static algorithms that falter in complex or deceptive environments. In such scenarios, human guidance can serve as a pivotal optimization mechanism, enhancing the rationality and safety of local target selection.

In the method proposed in this paper, human experience is first integrated into the learning process through a weighted control allocation strategy, enabling a smoother and more adaptive form of human intervention compared to traditional substitution-based approaches. Additionally, a set of local target points (SLTP) is extracted from real-time sensor data. An evaluation strategy is then applied to assess each candidate point, and the optimal LTP is selected. This process decomposes long-distance navigation tasks into multiple shorter sub-tasks, effectively alleviating the local optima problem. Furthermore, human dynamic guidance is incorporated into the LTP selection process, allowing humans to intervene in real time and provide predictive insights during navigation. Experimental results demonstrate that the proposed approach significantly improves the navigation performance of DRL-based reactive navigation in complex and unknown environments. The overall structure of this paper is illustrated in Fig. 1, and the main contributions are summarized as follows:

(1) We propose a human–machine deep reinforcement learning (HM-DRL) method for autonomous navigation in unknown environments, which achieves the dual-stage integration of human experience across both training and decision-making phases. Experimental results demonstrate that this approach effectively mitigates local optimality issues, boosting the navigation success rate from 77.58% (traditional DRL) to 93.26%.
(2) A human–machine co-decision mechanism (H-MCDM) is developed to dynamically fuse TD3-generated machine actions (a_m) with human expert inputs (a_h) via a weighted coefficient ω_m. By integrating this fused execution with experience replay, the method effectively overcomes the limitations of rigid action substitution and the poor adaptability of static demonstration buffers, enabling smoother control transitions and more efficient utilization of human guidance.
(3) A long-distance task decomposition mechanism based on LTPs is constructed, along with corresponding selection rules and evaluation metrics. By incorporating dynamic human guidance, the predictive capabilities of human operators are integrated into the LTP selection process. Experimental results show that this mechanism improves navigation success rates and alleviates local optimality problems.

The structure of this paper is arranged as follows: Section 2 presents the problem description and related research overview; Section 3 provides a detailed description of the method and structural design; Section 4 showcases experimental results and analysis; Section 5 concludes the paper and outlines future research directions.

2. Problem description and related work

2.1. Problem description

Existing DRL-based reactive navigation methods for complex, unknown environments face challenges in global optimization. Human–machine hybrid intelligence has emerged as an effective solution. Our model is described [35] by Eq. (1).

$$\max_{a_t \in A} J(s(t), a(t)) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s(k), a(k)) \tag{1a}$$

$$\text{s.t.} \quad a(t) = f^{a}(s(t), a_m(t), a_h(t), x(t)) \tag{1b}$$

$$a_m(t) = \pi(s(t)) \tag{1c}$$

$$a_h(t) = \text{Human-Action} \tag{1d}$$

$$s(t+1) = f^d(s(t), a(t)) \tag{1e}$$

$$C(s(t), a_m(t), a_h(t)) \leq 0 \tag{1f}$$

$$t = 0, 1, 2, 3, \ldots$$

Our objective function $J(s(t), a(t))$ aims to maximize the cumulative rewards over the course of the task, specifically defined as $J(s(t), a(t)) = \sum_{k=t}^{\infty} \gamma^{k-t} r(s(k), a(k))$, where $r(s(k), a(k))$ is the immediate reward at time step k, and the discount factor $\gamma \in [0, 1]$ accounts for the diminishing importance of future rewards. At each time step t, the system evaluates the current environmental state $s(t)$ and selects an action $a(t)$ using the action determination function $f^a(s(t), a_m(t), a_h(t), x(t))$. Here, $a_m(t)$ represents the machine-driven action derived from the robot's policy $\pi(s(t))$, where $\pi : \mathcal{S} \rightarrow \mathcal{A}$ denotes the mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$ (i.e., $a_m(t) = \pi(s(t))$), while $a_h(t) \in \mathcal{A}$ represents the human action component. These actions are combined using the arbitration function $x(t)$, balancing human and machine contributions to decision-making. The Markov property ensures that the state transition function $f^d(s(t), a(t))$ updates the environment's state as $s(t+1) = f^d(s(t), a(t))$, based solely on the current state $s(t)$ and the selected action $a(t)$. Additionally, the system operates under constraints $C(s(t), a_m(t), a_h(t)) \leq 0$, which ensure the feasibility of states and actions, such as avoiding obstacles or staying within operational boundaries.

2.2. Related work

2.2.1. Autonomous navigation in complex and unknown environments

Early autonomous navigation methods primarily relied on heuristic algorithms and rule-based navigation strategies, such as frontier-based exploration and heuristic path planning (e.g., A*, D*, and PRM) [4–6]. In recent years, the application of DRL in autonomous navigation has gained significant attention. Raja et al. optimized AI-driven aerial networks by combining DDPG with a leader–follower approach, enhancing decision-making performance [36]; Proximal Policy Optimization (PPO), stable in complex continuous spaces, is widely used in robotic and UAV navigation for obstacle avoidance and path planning [37]. Sensor fusion is also highlighted: Meng and Hsu proposed a resilient sensor fusion method for handling failures and uncertainties [38], while Zhu and Hayashibe designed a transfer learning-integrated hierarchical DRL framework to boost navigation efficiency and generalization [39].

Despite these advancements, the local optima problem remains a critical bottleneck in autonomous navigation, especially for long-distance tasks in complex scenarios, severely restricting navigation success and efficiency. One direction focuses on optimizing algorithms through reward shaping or safety control strategies: Han et al. enhanced collision avoidance capabilities via self-state-attention and sensor fusion, but their method only optimizes local perception and lacks global path awareness [17]; Miranda et al. improved the generalization of navigation systems through reward shaping, yet this approach relies heavily on manual rules and fails to adapt to dynamic obstacles [18]; Emam et al. ensured navigation safety using control barrier functions, but the conservative strategy sacrifices navigation efficiency and cannot decompose long-distance tasks [19]; it is worth noting that Shahid et al. focused on robotic manipulation control, which is irrelevant to navigation needs [20]. The other direction has proven effective by introducing LTP selection for task decomposition, which splits long and complex navigation tasks into multiple short and manageable subtasks to mitigate local optima. Cakmak et al. used intermediate waypoints for MAV navigation to split long paths into sub-tasks, significantly boosting task completion rates [31]; Xue et al. generated LTPs via maximum-entropy DRL, effectively preventing UAVs from falling into local optima [32,33]; Dong et al. integrated waypoint-based task decomposition to improve navigation success rate and path smoothness [34]. Although these methods have made progress, existing solutions still have limitations such as insufficient integration of human experience or lack of dynamic adaptability, leaving room for further optimization.

2.2.2. Human–machine hybrid intelligence

Existing human–machine hybrid methods typically integrate human experience at either the training or the decision-making stage, rather than both. During training, current approaches often employ replay buffers to store human demonstrations. For example, Luo et al. stored expert trajectories in static replay pools to guide policy updates [28]. However, the static nature of such buffers limits adaptability, as human experience collected in early training may not match later environmental changes. Other methods directly substitute human actions for machine decisions; for instance, Sun et al. and Wu et al. used human-adjusted parameters as auxiliary data [23,29], while Huang et al.'s "human as AI mentor" framework allows human actions to override the model when unsafe [30]. Yet, direct substitution is rigid and disrupts learning continuity in continuous action spaces, since training must pause for human input, slowing policy convergence and reducing overall efficiency.

At the decision-making stage, most research focuses on optimizing real-time human intervention, particularly in autonomous driving. Huang et al. proposed a shared-control framework where humans take over when machine control fails [40]. Gil et al. and Lian et al. monitored driver fatigue using EEG and hybrid EEG–eye-tracking methods to adjust intervention timing [41,42]. Wu et al. leveraged human guidance to bridge the "simulation-to-reality" gap in navigation [27]. Although these methods improve real-time adaptability, they generally lack the integration of expert knowledge during training, leaving the system unable to predict when human input is needed or how to adapt to human decisions. More critically, existing studies treat the training and decision-making stages as isolated processes, lacking a unified mechanism for cross-stage human–machine collaboration.

2.2.3. Deep reinforcement learning

In recent years, deep reinforcement learning has garnered significant attention in the field of autonomous navigation, with researchers proposing various improvements to address different challenges. Zhou et al. integrated prioritized experience replay (PER) with a double deep Q-network (DDQN) and introduced the blocking and blind angle (BBA) mechanism to mitigate blind spots in indoor exploration [43]. Li et al. combined deep reinforcement learning with the artificial potential field method, refining the action space and reward function of the DQN to enhance obstacle avoidance in path planning [44]. Liu et al. incorporated attention mechanisms into deep reinforcement learning algorithms to improve adaptability in real-world applications [45]. Lian et al. proposed a transferability metric based on scene similarity to evaluate the effectiveness of deploying models trained in simulations to real-world environments [46]. Jin et al. introduced the VWP autonomous navigation model, which enhances task completion rates in harsh environments [47]. Wu et al. adopted a dual-source training strategy to improve the success rate of deep reinforcement learning-based navigation [48]. Zhang et al. leveraged environmental information preprocessing to enhance the utilization of short-range LiDAR data, thereby improving navigation generalization [49]. Li et al. enhanced spatiotemporal reasoning to increase the interpretability of deep reinforcement learning in autonomous navigation tasks [50]. Collectively, these studies have advanced the application of deep reinforcement learning in autonomous navigation, offering novel solutions for efficient navigation in complex environments.

3. Methodology

The following subsections provide a detailed explanation of the two main components of this method: the H-MCDM and dynamic guidance based on local target points. These components realize a human–machine hybrid DRL that operates across both training and decision-making phases, thereby enhancing the long-distance autonomous navigation capability of mobile robots in complex unknown environments.

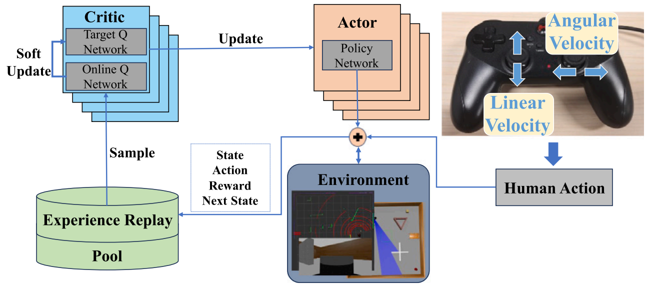


Fig. 2. The overall framework of the H-MCDM model. Human actions are input through the control handle and are weighted with the machine actions output by the actor network. These actions are applied to the environment to receive rewards, which are then stored in the sample pool. The critic network continuously updates its parameters through soft updates and optimizes the actor parameters using the TD error.

3.1. Human–machine co-decision mechanism

Although DRL can optimize decision-making by continuously learning from data in most cases, it may struggle to make globally optimal decisions when dealing with sudden events or unfamiliar environments due to delays or insufficient adaptability. In such situations, H-MCDM can incorporate human experience, enhancing the robot’s decision-making accuracy in unknown environments. The system architecture for H-MCDM is shown in Fig. 2. During the model training process, the action weights of the model and humans are allocated, and human experience and knowledge are directly integrated into the robot’s decision-making process. This is further reinforced through experience replay, compensating for the limitations of DRL in complex and unknown environments. The twin delayed deep deterministic policy gradient (TD3) algorithm is chosen as the framework to train this action strategy, addressing decision-making in continuous action spaces. The TD3 algorithm is an actor–critic model that uses separate actor and critic networks to generate and evaluate actions. As shown in Fig. 3, the TD3 network used in this paper consists of an actor network with three fully connected (FC) layers. The first two layers are followed by rectified linear unit (ReLU) activation functions, while the final layer employs a tanh activation function to constrain the action values. The state input to the actor network consists of 24 dimensions, which include both the environment and the robot’s own states. The network outputs linear velocity a_1 and angular velocity a_2. To comply with physical constraints in the real world, these outputs are scaled according to the maximum linear velocity v_{max} and angular velocity ω_{max}. Since the laser rangefinder’s data collection is limited to a 180-degree forward field of view, the linear velocity v is always set to non-negative values, ensuring the robot does not move backward. The final action values are computed as follows:

$$a = \left[v_{max} \left(\frac{a_1 + 1}{2} \right), \omega_{max} a_2 \right]. \tag{2}$$

The value of the state–action pair $Q(s, a)$ is evaluated by two critical networks. These two networks share the same architecture, but their parameter update strategies are asynchronous to allow for divergence in parameter values. Specifically, the state s and action a are input into the critic networks. The state s first passes through a fully connected layer, followed by a ReLU activation function, which outputs L_s. The output of this layer, along with the action a, is then fed into two independent transformation fully connected (TFC) layers, which are denoted as T_s and T_a, respectively. These vectors are combined with the output of a combination fully connected (CFC) layer L_c, where W_{T_s} and W_{T_a} represent the weights of T_s and T_a, and b_{T_a} is the bias of the T_a layer. The output of the combination layer, after passing through a ReLU activation, is connected to a fully connected layer, forming the final output Q-value. To reduce overestimation of the state–action pair values, the smaller Q-value between the two critic networks is selected as the final critic output.

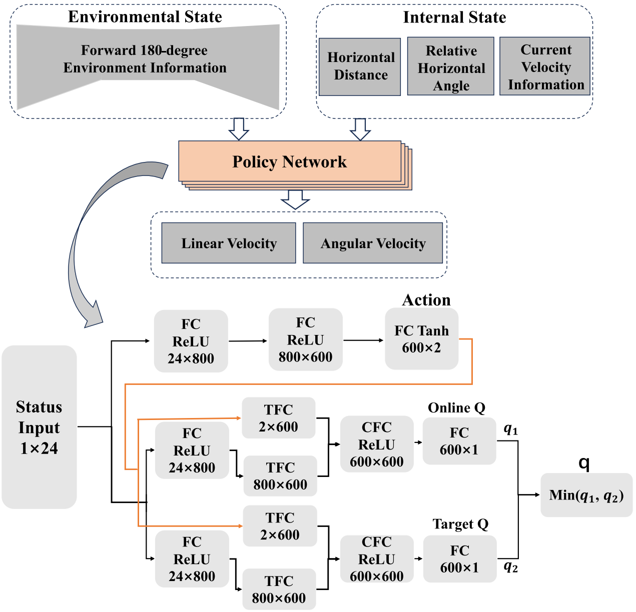


Fig. 3. The TD3 algorithm structure used in the H-MCDM model. The TD3 network consists of an actor and a critic. The network’s input includes the environmental state and the internal state of the agent, while the output is velocity information.

Algorithm 1 TD3 training process based on H-MCDM.

```
Initialize actor network $\pi_\theta$, critic networks $Q_{\phi_1}, Q_{\phi_2}$, and their target networks
Initialize replay buffer $\mathcal{B}$
for each training step do
    Select action with noise from actor network $a_h(t) = \pi_\theta(s_t) + \mathcal{N}_t$
    Execute action, store $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{B}$
    Sample a mini-batch from $\mathcal{B}$, compute target Q value:
    $y_t = r_t + \gamma \min(Q_{\phi'_1}(s_{t+1}), Q_{\phi'_2}(s_{t+1}))$
    Update critic networks $\phi_1, \phi_2$ by minimizing the loss:
    $L = (Q_{\phi_1}(s_t, a_t) - y_t)^2 + (Q_{\phi_2}(s_t, a_t) - y_t)^2$
    if policy update frequency reached then
        Update actor network $\theta$ by maximizing the Q value
        Soft update target networks:
        $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$
    end if
    Calculate mixed action with cooperative control strategy:
    $a_t = w_m \cdot a_m(t) + (1 - w_m) \cdot a_h(t)$
end for
```

$$L_c = L_s W_{T_s} + a W_{T_a} + b_{T_a}, \tag{3}$$

In each time step t, the reward r for a state–action pair (s_t, a_t) is determined by three conditions: if the distance d_g to the target at the current time step is less than a threshold δ_D, indicating that the target has been reached and the task is completed, a positive goal reward r_g is applied. If a collision is detected, a negative collision reward r_c is assigned. If neither of these conditions is met, the reward is based on the current linear velocity v, angular velocity ω, and the surrounding environmental context. By assigning weights to the linear and angular velocities, the system encourages the robot to move forward at a more stable pace. Specifically, a positive reward of $\frac{v}{2}$ is given for the linear velocity to encourage faster movement, while a negative reward of $-\frac{|w|}{2}$ is applied to the angular velocity to penalize sharp turns and avoid unnecessary trajectory deviations. This navigation strategy adopts a delayed reward optimization approach, targeting the goal with improved efficiency.

$$r(s_t, a_t) = \begin{cases} r_g & \text{if successful} \\ r_c & \text{if collision} \\ \frac{1}{2}(v - |w| - r_l(l_{min})) & \text{otherwise,} \end{cases} \tag{4}$$

The function $r_l(x)$ is a piecewise function designed to motivate the robot to maintain a safe distance from obstacles based on the minimum distance value l_{min} detected by the laser range sensor. By penalizing situations where the robot is too close to obstacles, the function further reinforces the robot's obstacle avoidance behavior. It is defined as:

$$r_l(x) = \begin{cases} 1 - x, & \text{if} x < 1 \\ 0, & \text{if} x \geq 1, \end{cases} \tag{5}$$

By assigning a weight ω_m, the human-in-the-loop control input can be softly integrated with the model output. Regarding human input methods, the trigger conditions are as follows: Human input is voluntary, meaning operators can intervene at any time during training. Additionally, the system provides supplementary prompts: when the model's actions lead to stagnation in local optima (i.e., the agent stops moving), the system sends a visual alert to encourage human guidance. The fused action is executed by the robot and simultaneously stored in the replay buffer. This process allows human experience to be incorporated into the DRL model through the experience replay mechanism. The model training process is shown in Algorithm 1. Regarding the value of weight ω_m — which represents the proportion of the DRL model's input in the fused action — it is adjusted based on environmental complexity to balance human input and model decisions, with a value range of $[0, 1]$. Specifically, ω_m reflects the priority of the DRL model: higher values indicate a greater emphasis on the model's output, while lower values mean the system prioritizes human input (via the weight $1 - \omega_m$). In simple environments, ω_m takes a higher value: the DRL model can execute tasks efficiently here, so relying more on its output reduces human input dependence and ensures speed. In complex or unknown environments, human input is particularly effective as it can predict future conditions, whereas the DRL model relies on past experiences. Thus, ω_m is decreased to prioritize human input (via an increased $1 - \omega_m$), leveraging its predictive capabilities to enhance adaptability.

3.2. Dynamic guidance based on local target points

The core concept of dynamic guidance based on local target points lies in decomposing long-path problems using LTP selection and evaluation. By integrating human predictive experience, the robot's navigation can be dynamically guided. The system framework for this approach is shown in Fig. 4. In this framework, the pre-trained strategy model combines LTP selection and evaluation for autonomous navigation. Human experience dynamically adjusts the robot's decisions by influencing the sequence of visiting these LTPs, indirectly affecting global path planning.

In the absence of prior knowledge about the environment, the robot must rely on real-time exploration to navigate towards the global target. This requires the robot not only to identify and move toward the destination but also to perceive and remember environmental features along the way, allowing it to quickly adjust its route when encountering obstacles or dead ends. During this process, the robot extracts a set of LTPs from the data received by its sensors and stores this information in memory. Specifically, the robot continuously scans its surroundings using its sensors to identify environmental features such as obstacles or corners, selecting appropriate points to add to SLTP in real time. When the robot's current distance from the global target exceeds a predefined threshold, or the uncertainty of directly completing the navigation task is high, a heuristic evaluation metric is applied to the points in SLTP to select the optimal point as the local target, guiding the robot toward the global target.

For marking points in SLTP, three situations are considered: When the laser sensor detects that there are no obstacles within a close-range sector of angle α and distance l_c, a point can be added on the bisector of this sector. This method can effectively guide the robot through the region, thereby advancing the task more quickly. Similarly, in a long-range sector of angle β and distance l_r, if the laser sensor detects no obstacles, a point can also be placed on the bisector of this sector to ensure continuity and smoothness of the navigation path. To further enhance navigation accuracy and flexibility, the system can analyze the difference between consecutive laser sensor readings. If the detected difference exceeds a predefined threshold d, it may indicate the presence of a gap through which the robot can pass. In such cases, a point can be added at the center of the gap, guiding the robot to utilize this potential passage for safe obstacle avoidance and path optimization. Fig. 5 illustrates the method of LTP selection and evaluation. Through this strategy, the robot can better adjust its navigation path using local environmental information, avoiding obstacles while preventing getting stuck in local optima.

This laser sensor-based LTP generation mechanism is highly practical in complex and unknown environments. When the global target is far in an unknown environment, directly using it as the autonomous navigation goal can easily lead to local optima. By accurately detecting environmental features and selecting appropriate LTPs, the robot can gradually approach the global target through successive visits to these local points. Once close to the global target, it can then be used as the driving goal for navigation. The stored SLTP is not fixed; when the robot visits a point, it is removed from the set. Additionally, if the robot fails to reach a selected LTP after multiple attempts, the point is also removed from the set. To determine the visitation order in the absence of human intervention, this paper proposes a heuristic evaluation method based on information distance limited exploration. Each point in the set is labeled as c_i, and the evaluation consists of three parts: the positional relationship between the robot's current location and the evaluation point (including position and angle), the positional relationship between the evaluation point and the global target, and the environmental information score of the evaluation point. The specific formula is as follows:

$$H_{C_i} = w_1 \cdot (d_{c_i} + |\beta_2|) + w_2 \cdot d_g + P_o \tag{6}$$

In the equation, d_{c_i} and d_g represent the Euclidean distance from the current position to the candidate point and the Euclidean distance from the candidate point to the global target, respectively. β_2 is the angle between the current heading and the candidate point, calculated via the inner product, as follows:

$$\beta_1 = \arccos\left(\frac{(x_c - x_o)}{\sqrt{(x_c - x_o)^2 + (y_c - y_o)^2}}\right) \tag{7}$$

$$\beta_2 = \beta_1 - \theta \tag{8}$$

The robot evaluates candidate points using a heuristic function that integrates distance, angle differences, and environmental information to select the optimal point as a local goal. Here, θ represents the current orientation, and β_2 denotes the angular difference between the direction of the candidate point and the robot. If $\beta_2 > \pi$ or $\beta_2 < -\pi$, it is adjusted to the range of $[-\pi, \pi]$. The forward cost consists of the distance from the robot to the candidate point and the angle difference, with w_1 and w_2 representing the weights for the forward and backward costs, respectively. Additionally, P_o assesses the obstacle situation around the candidate point. When there is an obstacle within a distance of 1 unit around the candidate point, P_o takes the value of -1;

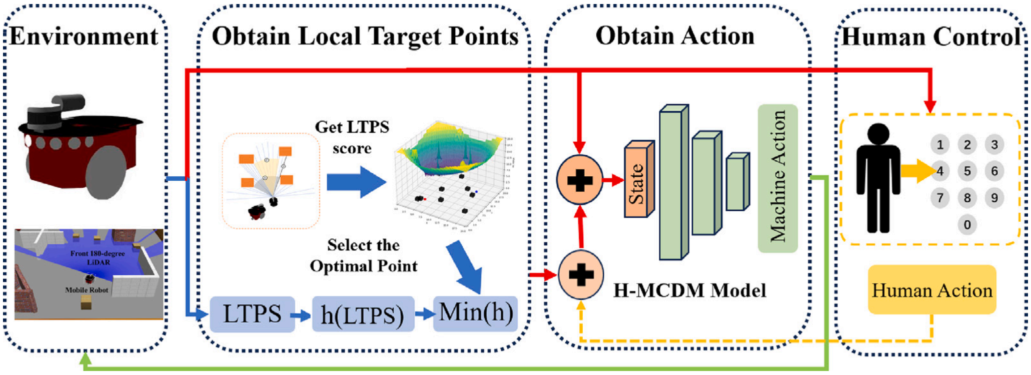


Fig. 4. Schematic diagram of the dynamic guidance based on local target points method. LTPs are obtained based on environmental information and serve as state inputs to the H-MCDM model network, generating the agent's action commands. Meanwhile, human control can dynamically guide the system by selecting LTPs from the SLTP using a button.

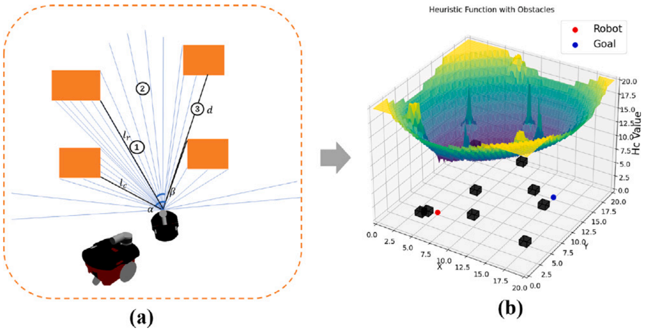


Fig. 5. This is the selection process of LTPs. (a) Selection of SLTP. The three numbered markers in the figure correspond to three different types of selection for SLTP. (b) The heuristic LTP evaluation method used, generating a three-dimensional image.

Algorithm 2 LTP Selection and Evaluation with Pre-trained H-MCDM Model.

```
Initialize pre-trained H-MCDM model $\mathcal{M}$, SLTP list $\mathcal{P}$, and
  evaluation weights $w_1$, $w_2$
for each step do
    Select action $a_m(t) = \mathcal{M}(s_t) + \mathcal{N}_t$, execute, store $(s_t, a_t, r_t, s_{t+1})$ in
  $\mathcal{B}$
    LTP Selection: Add LTPs to $\mathcal{P}$ if conditions are met
    LTP Evaluation:
    for each LTP $c_i \in \mathcal{P}$ do
        Compute heuristic score $H_{C_i}$
    end for
    Local Target:
    if $\lambda_h$ then
        LTP = Human-selected
    else
        LTP = $argmin(H_{C_i})$
    end if
    Remove visited/unreachable LTP from $\mathcal{P}$
    Action Update: Apply action and proceed
end for
```

otherwise, it takes the value of 0. This helps ensure effective collision avoidance.

In navigation tasks, a robot often needs to pass through multiple LTPs to reach its final destination. The selection of these intermediate points plays a critical role in determining the overall effectiveness and efficiency of the robot's path planning strategy. These LTPs represent strategic waypoints that guide the robot's movements, ensuring it follows an optimal trajectory that avoids obstacles, minimizes travel time, and maximizes safety. The process of selecting and evaluating LTPs, therefore, directly impacts the robot's ability to navigate through complex environments. In our method, the selection and evaluation of LTPs are based on a SLTP. The SLTP has a maximum capacity of 400, meaning up to 400 LTPs are compared per episode; in practice, the actual number of candidate LTPs per episode is far less than 400, confirming sufficient capacity. The SLTP is dynamically updated: visited LTPs are removed, and LTPs that the robot fails to reach after multiple attempts are also eliminated. Humans play a crucial role in adjusting the LTPs based on their experience and ability to make predictions about future situations. Specifically, the top 10 LTPs with the highest H_{C_i} scores are displayed in the RViz visualization interface, numbered 0–9, for human reference; operators can select an LTP by entering the corresponding number 0–9 via a standard keyboard, with no confirmation key required and the input taking effect immediately in the next decision step. When sensor data is incomplete, or when there are uncertainties about the environment, such human input can provide corrective guidance.

The detailed process of this method, including the steps of LTP selection, evaluation, and adjustment, is shown in Algorithm 2. λ_h is a Boolean variable that equals 1 when human involvement is present in LTP selection (i.e., when operators input an LTP number as described), and 0 otherwise. This algorithm captures the integration of both human and machine decision-making in the context of robot navigation. By dynamically adjusting the LTPs, the robot's global optimization capability is significantly improved, particularly in complex and unknown environments.

4. Experiments

The overall objective of the experiments in this section is to systematically validate the effectiveness of the proposed HM-DRL method for autonomous navigation, with targeted verification of its three core components: (1) Verify whether the H-MCDM can integrate human expertise with DRL's self-learning capabilities, thereby improving navigation performance in unknown environments; (2) Confirm whether the LTP selection and evaluation strategy can decompose long-distance tasks to mitigate the local optima problem of reactive navigation; (3) Validate whether integrating human predictive capabilities into LTP selection can further enhance global optimization ability, especially in scenarios with high obstacle density or sudden environmental changes.

To achieve the above objectives, this section provides a detailed overview of the experimental design and implementation process. First, we outline the specific experimental settings. Subsequently, we validate the navigation performance improvement brought by H-MCDM (Section 4.2). Building on this, we verify the role of LTPs and human dynamic guidance in global optimization (Section 4.3).

Table 1
Parameter settings.

Parameter	Value
Learning rate	0.001
Soft update coefficient τ	0.05
Standard Gaussian noise $\mathcal{N}_t$	[−0.2, 0.2]
Update frequency for policy network	2
Max steps per episode	500
Target reward r_g	100
Collision r_c	−100
Parameters in SLTP selection α, β	$\pi/3, \pi/6$
Parameters in SLTP selection l_c, l_r, d	$1.5, 2.5, 1.5$
Maximum storage capacity of SLTP	400
LTP evaluation weight parameterω_1, ω_2	1.3, 0.7
Hybrid weight ω_m	0.5

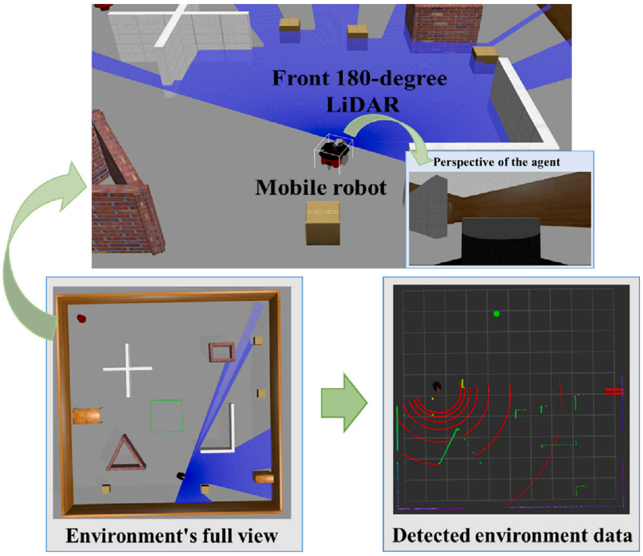


Fig. 6. Autonomous navigation environment in Gazebo and the visual map of environmental information detected by LiDAR.

4.1. Experimental setup

This section details the unified experimental setup, which is designed to provide a consistent, reproducible baseline for verifying the proposed method—ensuring that differences in experimental results can be attributed to the core components of HM-DRL rather than random factors.

The experiments in this study were conducted using the Gazebo simulator to validate the effectiveness of the proposed navigation strategy. The training environment is a simulated 10 × 10 meter area, as shown in Fig. 6, which contains various obstacle shapes and randomly generated obstacles. Each training session concludes when one of the following conditions is met: the robot successfully reaches the target point, a collision occurs, or the maximum step limit of 500 is exceeded. During training, the robot's maximum linear velocity is set to 1 m/s, and its maximum angular velocity is limited to 1 rad/s. Movement commands are issued with a delay t_w to simulate communication latency. Gaussian noise, denoted as $\mathcal{N}_t$, is added to both sensor inputs and action outputs. The robot's initial position, target location, and obstacle positions are randomized in each training session. The robot is equipped with an RpLidar sensor, which has a 10-meter range and is divided into 20 sectors, with the minimum value from each sector used as the input state. The robot is considered to have reached the target when it is within 0.3 m, and a collision is detected when an obstacle is within 0.35 m. Negative rewards are assigned if the robot comes too close to an obstacle, and training ends upon collision. The robot's orientation is represented by quaternions, which are then converted to Euler angles to ensure accurate posture tracking during navigation.

Key parameters for the proposed navigation strategy are summarized in Table 1. These include the learning rate, soft update coefficient, noise range, policy network update frequency, and other critical settings to ensure effective training and evaluation.

During the training process using H-MCDM, the robot's initial and target positions, as well as obstacle distributions, are randomized to enhance data diversity, and Gaussian noise is introduced to simulate real-world uncertainties. A target network stabilizes training by periodically updating its parameters, using a soft update mechanism with a ratio of $\tau = 0.005$, which smooths parameter updates and prevents large fluctuations. The critic is optimized using temporal difference (TD) error, while the actor is trained by maximizing the negative Q-value. It is important to integrate human expertise and predictions into the machine model. In simple environments where the robot can complete tasks independently, human intervention is not needed, and the machine's action a_m is the final output. However, in complex environments, human intervention is required to combine the human action a_h with the machine action a_m. Human operators utilize a standard BTP-2270U gamepad for control. The vertical displacement of the left joystick is mapped to the linear velocity, v_h, where the neutral position corresponds to 0 and a full upward push corresponds to 1. The horizontal displacement of the right joystick maps to the angular velocity, ω_h, where a full left push corresponds to −1, a full right push corresponds to 1, and the neutral position corresponds to 0. Fig. 7(a) and (b) show the relationship between the linear and angular velocities of human and machine actions when $\omega_m = 0.5$. To align with our initial settings, the human's linear velocity ranges from [0, 1], and the angular velocity ranges from [−1, 1].

4.2. Experiment 1: human–machine co-decision-making strategy

The specific objective of this experiment is to verify the effectiveness of the H-MCDM. To achieve this, we first train a TD3 model using the H-MCDM mechanism—monitoring metrics such as episode rewards, maximum Q-values, and average Q-values estimated by the critic network to ensure the model converges to a stable, usable state. On this basis, we compare the navigation performance of this H-MCDM-trained model with that of traditional DRL models. The ultimate goal is to confirm whether H-MCDM can reduce collision frequencies and enhance task success rates in complex, unknown navigation environments.

The TD3 model is trained using H-MCDM. As shown in Fig. 8(a), the changes in rewards per episode during training are illustrated. The figure indicates that the rewards continuously increase from episode 0 to around episode 4000, and after episode 4000, the rewards gradually stabilize, signaling the completion of the model's training. Fig. 8(b) and (c) present the maximum Q-values and average Q-values of the state–action pairs at a specific timestep during pre-training. The maximum Q-value, estimated by the critic-target network, represents the highest cumulative reward the agent can achieve by taking the optimal action in the current state. The average Q-value reflects the overall trend of the agent's strategy performance throughout the training process. As seen in the figure, the average Q-value gradually increases, indicating that the agent is learning to achieve higher rewards, with its strategy being continuously optimized until reaching an optimal level.

After completing the training of the robot model, we evaluated the effectiveness of the control authority coordination strategy by comparing the robot's navigation performance in complex, unknown environments under both the traditional DRL model and H-MCDM model. We analyzed the navigation performance from multiple perspectives, including rewards per episode, the number of successes and failures per epoch, average rewards, and overall task success rate. For the comparison of navigation performance, four different models were prepared: three models were trained using the H-MCDM method with

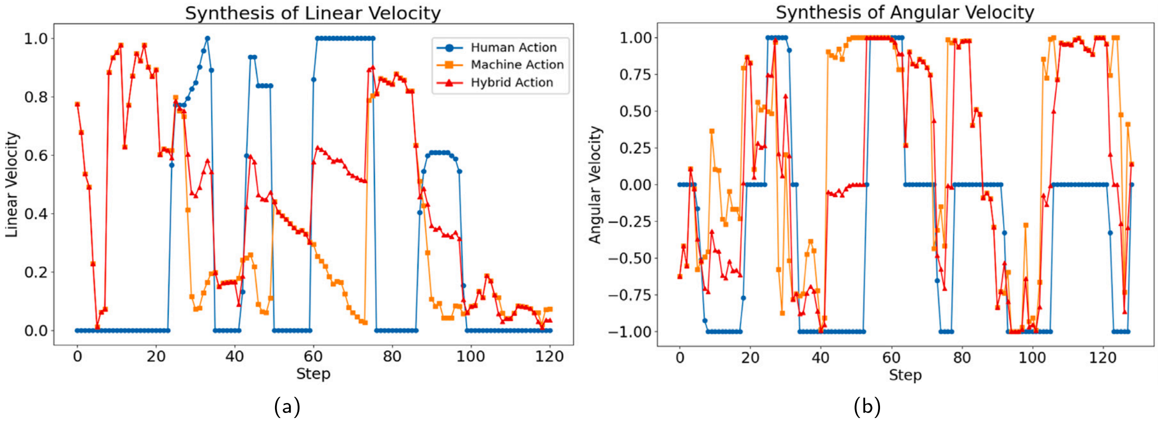


Fig. 7. This shows the machine actions and human actions during the H-MCDM model training process when $\omega_m = 0.5$, where (a) represents the hybrid of linear velocity and (b) represents the hybrid of angular velocity.

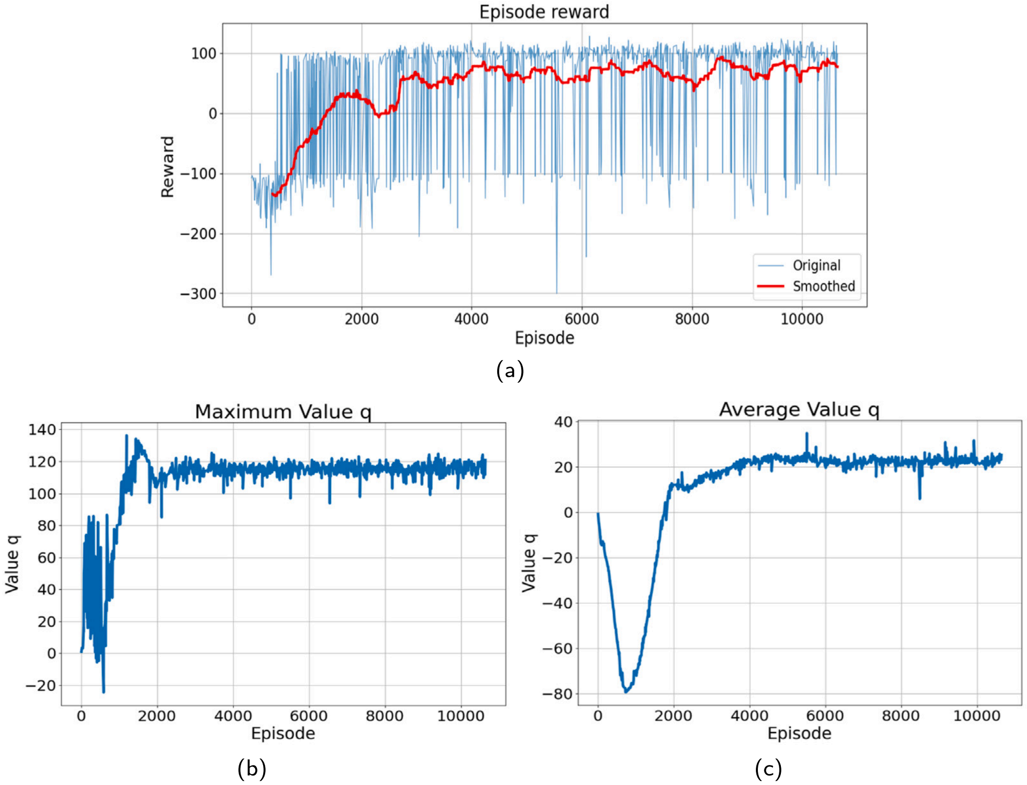


Fig. 8. (a) The variation in rewards during the training process. (b) The maximum Q-value estimated by the Critic network during training. (c) The average Q-value estimated by the critic network during training.

10,000, 500, and 100 episodes, named H-MCDM, Model-100ep, and Model-500ep respectively. One model was trained using the basic DRL method with 10,000 training episodes, named Basic-DRL. In addition, we investigated the navigation performance of pure human strategies in unknown environments. The experiment adopted joystick control for input, and to ensure that humans and the machine received the same environmental information, operators could only view the environmental information detected by the robot via the RViz visualization interface. This strategy is named Human.

To ensure the objectivity and validity of the experiments while avoiding random errors, a total of 100 epochs of tasks were conducted. Each epoch contained 50 episodes, resulting in a total of 5000 navigation tasks. Since pure human experiments require a considerable amount of time, we conducted 500 episodes. These results are represented by the yellow curve in the partial subplot of Fig. 9. Episode reward quantifies the comprehensive efficiency of a single navigation episode, integrating path optimality, task completion speed, and collision penalties. Calculated by the predefined reward function: +100 for target arrival, −100 for collisions, and negative penalties for obstacle proximity. Higher values mean more optimized navigation. Rewards obtained in each episode were recorded, and to facilitate the observation of the experimental results, we applied the exponential moving average (EMA) technique for smoothing. These results are shown in Fig. 9. The Model-100ep and Model-500ep models, due to insufficient training, exhibited rewards significantly lower than the Basic-DRL model. The H-MCDM model, benefiting from human experience, showed better navigation performance, with its reward curve positioned above that of the Basic-DRL and the pure human strategy named Human.

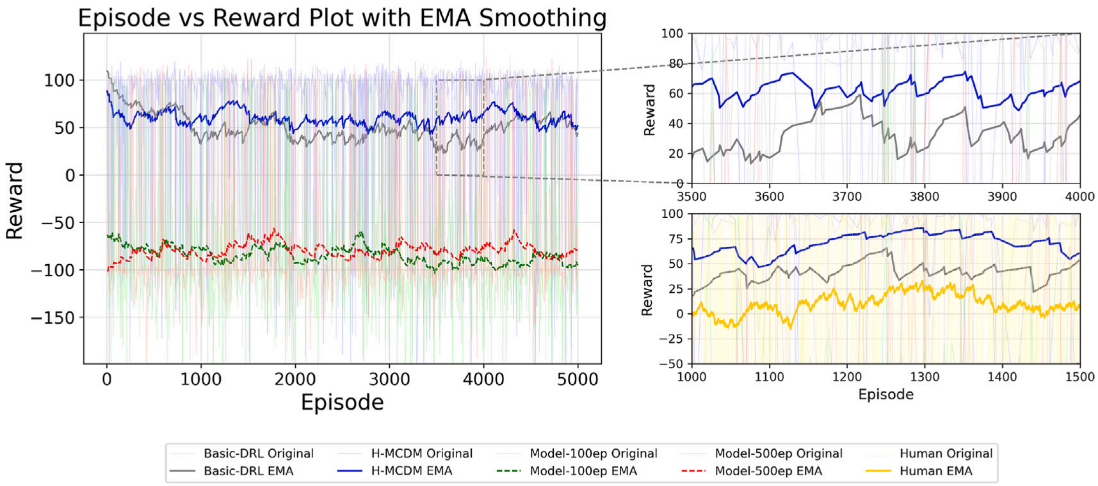
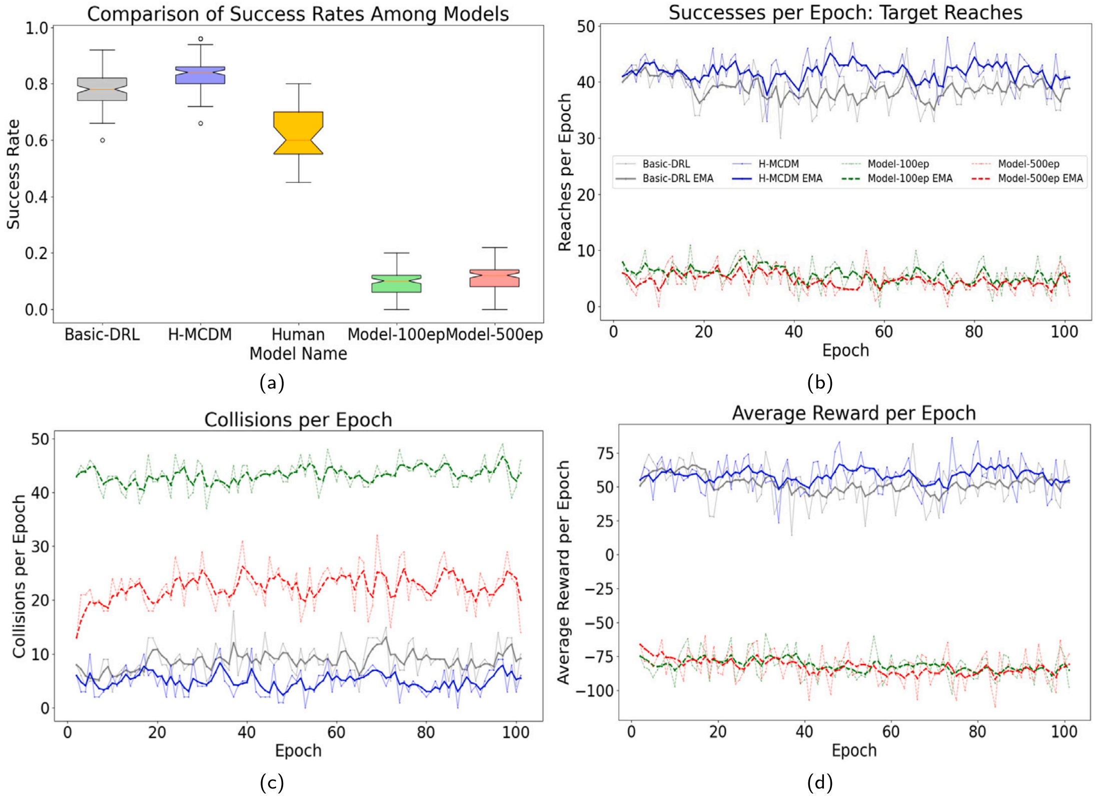


Fig. 9. Comparison of the rewards for each episode between Basic-DRL, H-MCDM, and Human.



(a)

(b)

(c)

(d)

Fig. 10. Comparison of metrics between different models. (a) Box plot of success rates for different models; (b) Comparison of the number of successes per epoch for different models; (c) Comparison of the number of collisions per epoch for different models; (d) Comparison of the average reward per epoch for different models.

Specifically, the success rate for each epoch was calculated as the ratio of the number of episodes reaching the target point to the total number of episodes. The box plot in Fig. 10(a) depicts the distribution of these epoch-wise success rates. Additionally, it should be noted that the pure human operation Human is limited to 500 episodes due to time constraints, and thus its results were divided into 25 epochs each comprising 20 episodes, while the remaining models were divided into 100 epochs each with 50 episodes. With the progress of training, the navigation task success rate of the Model-500ep was slightly higher than that of the Model-100ep, but both were significantly lower than that of the Basic-DRL (77.58%). For the Human, 311 out of 500 training episodes were successfully completed, corresponding to a success rate of 62.2%, while the H-MCDM model achieved a success rate of 83.5%, which was significantly higher than those of the Basic-DRL and the pure human operation. To further investigate the navigation performance of the four models, we analyzed each epoch during the experiment. For each epoch, the task success count, collision count, and average reward for the 50 episodes were recorded, as shown in Fig. 10. As can be seen from Fig. 10(b), the Basic-DRL model had a significantly higher success count in each epoch compared to the Model-100ep and Model-500ep models, but it was still lower than that of the H-MCDM model. In Fig. 10(c), it is evident that the H-MCDM model had the fewest collisions, while the Model-500ep model experienced far fewer collisions than the Model-100ep model. This is because, as the number of training episodes

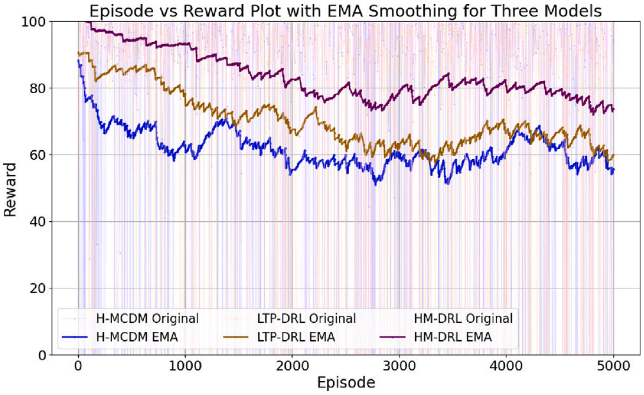


Fig. 11. Comparison of rewards per episode for the H-MCDM, LTP-DRL, and HM-DRL models.

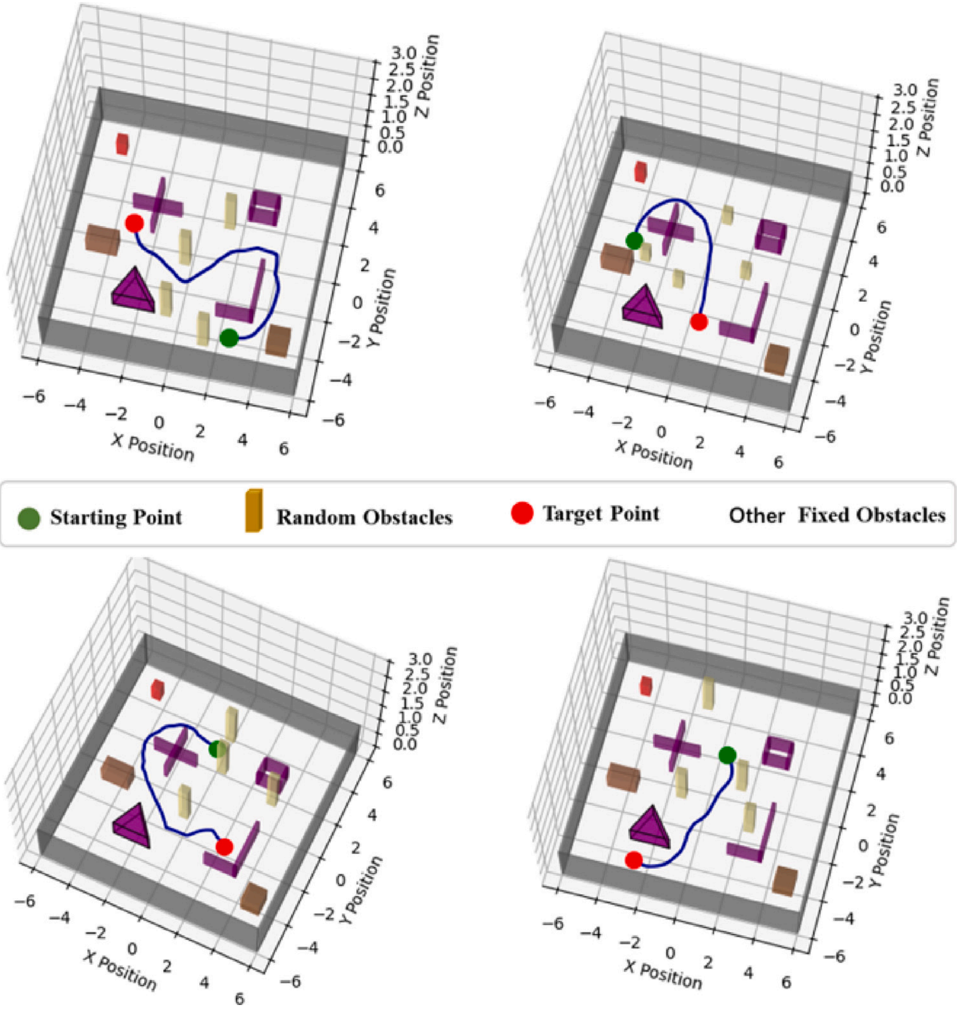
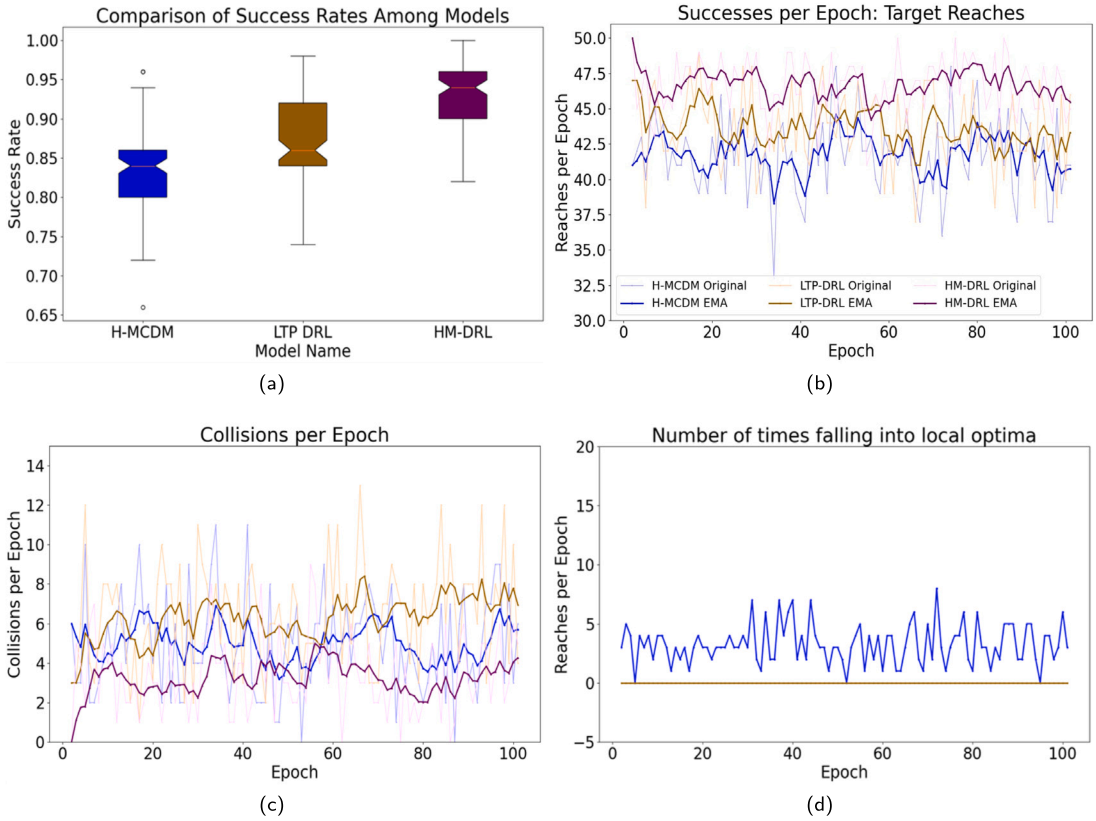


Fig. 12. Path trajectory of the agent in the simulation environment using the HM-DRL model.

increased, the model gradually realized that collisions resulted in significant penalties, and thus made efforts to avoid collisions. However, due to insufficient training, the model was still unable to complete tasks successfully and fell into local optima. Similarly, in Fig. 10(d), it is clear that the H-MCDM model outperforms the Basic-DRL model in terms of navigation performance. This analysis of the experimental process further demonstrates that human experience contributes to an improvement in navigation performance.

Experimental results demonstrate that the H-MCDM model significantly outperforms both the Basic-DRL method and the pure human model Human in key evaluation metrics, including navigation task rewards and task success rate. This finding validates that compared with relying solely on intelligent algorithms or pure human operation, the human–machine hybrid yields more stable and optimized navigation outcomes.



(a)

(b)

(c)

(d)

Fig. 13. Comparison of metrics between the H-MCDM, LTP-DRL, and HM-DRL accident models. (a) Box plot of success rates for different models; (b) Comparison of the number of successes per epoch for different models; (c) Comparison of the number of collisions per epoch for different models; (d) Comparison of the number of times the models get trapped in local optima per epoch.

4.3. Experiment 2: dynamic path guidance under local target points

The specific objective of this experiment is twofold. First, it aims to verify whether the LTP selection and evaluation strategy can decompose long-distance tasks and thereby mitigate the local optima problem. This verification will be conducted by comparing the performance of a model integrated with LTP against the baseline H-MCDM model from Section 4.2. Second, it seeks to confirm whether integrating human dynamic guidance into the LTP selection process can further enhance the system's global optimization ability. This confirmation will be realized by comparing the proposed HM-DRL method against the model that only includes LTP; the HM-DRL method combines both LTP and human guidance.

To achieve these two objectives, this experiment directly builds on the H-MCDM model established in Section 4.2. We first introduce local target points to segment long-distance navigation paths, which results in a modified model named LTP-DRL. On this basis, we further incorporate dynamic human guidance into the LTP selection process of LTP-DRL to form the final proposed HM-DRL method. This method is used to verify the additional value of human intervention. To minimize experimental bias caused by random factors, the experiment adopts the identical setup as Section 4.2. This setup includes 100 epochs of testing, 50 episodes per epoch, and consistent environmental parameters such as obstacle distribution rules and robot kinematic constraints.

Similarly, we recorded the rewards per episode for the H-MCDM, LTP-DRL, and HM-DRL models in a complex, unknown environment. To facilitate performance comparison, we applied EMA smoothing to the data, and the results are shown in Fig. 11. From the displayed rewards, it can be observed that the LTP-DRL model, which incorporates path segmentation, outperforms the H-MCDM model for most of the time, but it is still below the HM-DRL model, which combines both path segmentation and human predictions. To qualitatively analyze the performance of the HM-DRL method, we saved the autonomous navigation trajectories using the HM-DRL model and plotted the path trajectory shown in Fig. 12. This includes the start point, target point, a variety of fixed obstacles, random obstacles, and the navigation path. From the autonomous navigation paths, it can be observed that the HM-DRL model dynamically adjusts its commands based on the surrounding environmental conditions, successfully completing the task.

For quantitative analysis of the model's performance, we analyzed each epoch of the experiment, recording the number of successes, collisions, and instances of local optima for each epoch of 50 episodes. The recorded results are plotted in Fig. 13. The success rates from the 5000-episode experiment are summarized in the box plot shown in Fig. 13(a). The success rates for the H-MCDM (83.5%), LTP-DRL (87.14%), and HM-DRL (93.26%) models increase sequentially, indicating that both path segmentation using LTP and dynamic guidance through human prediction improve the navigation performance of the model. The model combining both methods, HM-DRL, achieves the best navigation performance. To facilitate comparison, the success and collision counts, which exhibited large variability, were smoothed using EMA. From Fig. 13(b), it can be seen that the number of successes per epoch for the HM-DRL, LTP-DRL, and H-MCDM models shows a decreasing trend in sequence. In Fig. 13(c), the collision counts for the LTP-DRL and H-MCDM models are quite similar, with the former slightly higher, while HM-DRL has the fewest collisions. This is because LTP-DRL failed to properly handle some local optimal situations. If the robot's euclidean displacement is less than 0.2 for 50 continuous steps, it is deemed to be in a local optimum. Fig. 13(d) shows the number of times the model falls into local optima during each epoch. Here, the curves for the HM-DRL and LTP-DRL models overlap and are both zero, while the H-MCDM model lies above them.

Through the qualitative and quantitative analysis of the above experiments, it is evident that incorporating LTP to decompose the navigation path helps mitigate the issue of local optima commonly encountered in reactive navigation, thus improving overall navigation performance. The dynamic guidance provided by human predictions enhances the model by adjusting the sequence in which LTPs are accessed, effectively integrating human foresight of future conditions into the navigation process. This approach significantly boosts the robot's navigation performance, particularly in complex and unknown environments.

5. Conclusion and future work

This paper proposes an HM-DRL autonomous navigation system that integrates human experiential knowledge with a heuristic LTP evaluation mechanism to address challenges in complex and unknown environments. By overcoming the local optimum trap and performance limitations of traditional DRL-based reactive navigation, the system improves task success rates, reduces collisions, and enhances overall navigation performance. Experimental results validate the effectiveness of human–machine collaboration, particularly in incorporating human predictive capabilities through dynamic guidance. However, challenges remain, including optimizing control authority allocation for better human–machine collaboration, enhancing the universality of the LTP evaluation strategy, and exploring online learning mechanisms to enable autonomous optimization of human experience, reducing reliance on human intervention and improving system adaptability and autonomy.

In future work, we aim to address the remaining challenges by focusing on two key areas. First, we will develop advanced control authority allocation strategies to optimize human–machine collaboration, ensuring seamless interaction and improved system efficiency. Second, we plan to enhance the universality of the LTP evaluation mechanism, such as surface and underwater robots and aircraft, making it more robust and adaptable to diverse environments. These advancements will further strengthen the system's autonomy and performance in complex and dynamic scenarios.

CRediT authorship contribution statement

Yongheng Li: Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Qianqian Zhang:** Writing – original draft, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Yu Kang:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Yun-Bo Zhao:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62203006, Grant No. 62173317).

Data availability

Data will be made available on request.

References

[1] H. Jing, Y. Gao, S. Shahbeigi, M. Dianati, Integrity monitoring of gnss/ins based positioning systems for autonomous vehicles: State-of-the-art and open challenges, IEEE Trans. Intell. Transp. Syst. 23 (2022) 14166–14187.

[2] D. Koung, O. Kermorgant, I. Fantoni, L. Belouaer, Cooperative multi-robot object transportation system based on hierarchical quadratic programming, IEEE Rob. Autom. Lett. 6 (2021) 6466–6472.

[3] Y. Mo, Z. Sun, C. Yu, Eventtube: An artificial intelligent edge computing based event aware system to collaborate with individual devices in logistics systems, IEEE Trans. Ind. Informat. 19 (2022) 1823–1832.

[4] M. Wang, J. Xu, J. Zhang, Y. Cui, An autonomous navigation method for orchard rows based on a combination of an improved a-star algorithm and svr, Precis. Agric. 25 (2024) 1429–1453.

[5] B. Sun, W. Zhang, S. Li, X. Zhu, Energy optimised d* auv path planning with obstacle avoidance and ocean current environment, J. Navig. 75 (2022) 685–703.

[6] A. Francis, A. Faust, H.-T. Chiang, J. Hsu, J. Kew, M. Fiser, T.-W. Lee, Long-range indoor navigation with prm-rl, IEEE Trans. Robot. 36 (2020) 1115–1134.

[7] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, et al., A review: On path planning strategies for navigation of mobile robot, Def. Technol. 15 (2019) 582–606.

[8] D. Lee, S. Lee, C. Ahn, P. Shi, C.-C. Lim, Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot, IEEE Trans. Ind. Electron. 68 (2020) 9998–10006.

[9] W. Yang, P. Wu, X. Zhou, H. Lv, X. Liu, G. Zhang, Z. Hou, W. Wang, Improved artificial potential field and dynamic window method for amphibious robot fish path planning, Appl. Sci. 11 (2021) 2114.

[10] P. Chen, J. Pei, W. Lu, M. Li, A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance, Neurocomputing 497 (2022) 64–75.

[11] Y. Wang, Y. Xie, D. Xu, J. Shi, S. Fang, W. Gui, Heuristic dense reward shaping for learning-based map-free navigation of industrial automatic mobile robots, ISA Trans. 156 (2024) 579–596.

[12] H. Li, Q. Zhang, D. Zhao, Deep reinforcement learning-based automatic exploration for navigation in unknown environment, IEEE Trans. Neural Netw. Learn. Syst. 31 (2019) 2064–2076.

[13] R. Cimurs, I. Suh, J. Lee, Goal-driven autonomous exploration through deep reinforcement learning, IEEE Robot. Autom. Lett. 7 (2021) 730–737.

[14] Z. Xu, B. Liu, X. Xiao, A. Nair, P. Stone, Benchmarking reinforcement learning techniques for autonomous navigation, 2023, pp. 9224–9230.

[15] J. Choi, K. Park, M. Kim, S. Seok, Deep reinforcement learning of navigation in a complex and crowded environment with a limited field of view, 2019, pp. 5993–6000.

[16] G. Chen, L. Pan, Y. Chen, P. Xu, Z. Wang, P. Wu, J. Ji, X. Chen, Deep reinforcement learning of map-based obstacle avoidance for mobile robot navigation, SN Comput. Sci. 2 (2021) 1–14.

[17] Y. Han, I. Zhan, W. Zhao, J. Pan, Z. Zhang, Y. Wang, Y.-J. Liu, Deep reinforcement learning for robot collision avoidance with self-state-attention and sensor fusion, IEEE Rob. Autom. Lett. 7 (2022) 6886–6893.

[18] V. Miranda, A. Neto, G. Freitas, L. Mozelli, Generalization in deep reinforcement learning for robotic navigation by reward shaping, IEEE Trans. Ind. Electron. 71 (2023) 6013–6020.

[19] Y. Emam, G. Notomista, P. Glotfelter, Z. Kira, M. Egerstedt, Safe reinforcement learning using robust control barrier functions, IEEE Rob. Autom. Lett. 10 (2022) 2886–2893.

[20] A. Shahid, D. Piga, F. Braghin, L. Roveda, Continuous control actions learning and adaptation for robotic manipulation through reinforcement learning, Auton. Robots 46 (2022) 483–498.

[21] Y. Yan, J. Wang, K. Zhang, Y. Liu, Y. Liu, G. Yin, Driver's individual risk perception-based trajectory planning: A human-like method, IEEE Trans. Intell. Transp. Syst. 23 (2022) 20413–20428.

[22] B. Patel, L. Rosenberg, G. Willcox, D. Baltaxe, M. Lyons, J. Irvin, P. Rajpurkar, T. Amrhein, R. Gupta, S. Halabi, et al., Human–machine partnership with artificial intelligence for chest radiograph diagnosis, NPJ Digit. Med. 2 (2019) 111.

[23] Y. Wu, L. Ma, X. Yuan, Q. Li, Human–machine hybrid intelligence for the generation of car frontal forms, Adv. Eng. Inf. 55 (2023) 101906.

[24] J. Ostheimer, S. Chowdhury, S. Iqbal, An alliance of humans and machines for machine learning: Hybrid intelligent systems and their design principles, Technol. Soc. 66 (2021) 101647.

[25] M. Vaccaro, A. Almaatouq, T. Malone, When combinations of humans and ai are useful: A systematic review and meta-analysis, Nat. Hum. Behav. 8 (2024) 2293–2303.

[26] J. Korteling, G. van de Boer-Visschedijk, R. Blankendaal, R. Boonekamp, A. Eikelboom, Human-versus artificial intelligence, Front. Artif. Intell. 4 (2021) 622364.

[27] J. Wu, Y. Zhou, H. Yang, Z. Huang, C. Lv, Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation, IEEE Trans. Pattern Anal. Mach. Intell. 45 (2023) 14745–14759.

[28] B. Luo, Z. Wu, F. Zhou, B.-C. Wang, Human-in-the-loop reinforcement learning in continuous-action space, IEEE Trans. Neural Netw. Learn. Syst. 35 (2023) 15735–15744.

[29] X. Sun, Z. Xu, J. Qiu, H. Liu, H. Wu, Y. Tao, Optimal volt/var control for unbalanced distribution networks with human-in-the-loop deep reinforcement learning, IEEE Trans. Smart Grid 15 (2023) 2639–2651.

[30] Z. Huang, Z. Sheng, C. Ma, S. Chen, Human as ai mentor: Enhanced human-in-the-loop reinforcement learning for safe and efficient autonomous driving, Commun. Transp. Res. 4 (2024) 100127.

[31] F. Cakmak, S. Yavuz, A 3d navigation algorithm switching between waypoint and bezier curves based local plans for micro air vehicles, Eng. Sci. Technol. Int. J. 48 (2023) 101560.

[32] Y. Xue, W. Chen, Combining motion planner and deep reinforcement learning for uav navigation in unknown environment, IEEE Robot. Autom. Lett. 9 (2023) 635–642.

[33] Y. Xue, W. Chen, Rloplanner: Combining learning and motion planner for uav safe navigation in cluttered unknown environments, IEEE Trans. Veh. Technol. 73 (2023) 4904–4917.

[34] T. Dong, X. Song, Y. Zhang, X. Qin, Y. Liu, Z. Bai, Vit-enabled task-driven autonomous heuristic navigation based on deep reinforcement learning, IEEE Robot. Autom. Lett. 10 (2025) 5297–5304.

[35] Q. Zhang, Y. Kang, Y.-B. Zhao, P. Li, S. You, Traded control of human–machine systems for sequential decision-making based on reinforcement learning, IEEE Trans. Artif. Intell. 3 (2021) 553–566.

[36] G. Raja, S. Essaky, A. Ganapathisubramaniyan, Y. Baskar, Nexus of deep reinforcement learning and leader–follower approach for aiot enabled aerial networks, IEEE Trans. Ind. Informat. 19 (2022) 9165–9172.

[37] A. Alagha, S. Singh, R. Mizouni, J. Bentahar, H. Otrok, Target localization using multi-agent deep reinforcement learning with proximal policy optimization, Fut. Gener. Comput. Syst. 136 (2022) 342–357.

[38] Q. Meng, L.-T. Hsu, Resilient interactive sensor-independent-update fusion navigation method, IEEE Trans. Intell. Transp. Syst. 23 (2022) 16433–16447.

[39] W. Zhu, M. Hayashibe, A hierarchical deep reinforcement learning framework with high efficiency and generalization for fast and safe navigation, IEEE Trans. Ind. Electron. 70 (2022) 4962–4971.

[40] W. Huang, H. Liu, Z. Huang, C. Lv, Safety-aware human-in-the-loop reinforcement learning with shared control for autonomous driving, IEEE Trans. Intell. Transp. Syst. 25 (2024) 16181–16192.

[41] M. Gil, M. Albert, J. Fons, V. Pelechano, Designing human-in-the-loop autonomous cyber-physical systems, Int. J. Hum.-Comput. Stud. 130 (2019) 21–39.

[42] Z. Lian, T. Xu, Z. Yuan, J. Li, N. Thakor, H. Wang, Driving fatigue detection based on hybrid electroencephalography and eye tracking, IEEE J. Biomed. Health Informat. 28 (2024) 6568–6580.

[43] Y. Zhou, J. Yang, Z. Guo, Y. Shen, K. Yu, J.-W. Lin, An indoor blind area-oriented autonomous robotic path planning approach using deep reinforcement learning, Expert Syst. Appl. 254 (2024) 124277.

[44] L. Li, D. Wu, Y. Huang, Z.-M. Yuan, A path planning strategy unified with a colregs collision avoidance function based on deep reinforcement learning and artificial potential field, Appl. Ocean Res. 113 (2021) 102759.

[45] Z. Liu, Y. Cao, J. Chen, J. Li, A hierarchical reinforcement learning algorithm based on attention mechanism for uav autonomous navigation, IEEE Trans. Intell. Transp. Syst 24 (2022) 13309–13320.

[46] S. Lian, F. Zhang, A transferability metric using scene similarity and local map observation for drl navigation, IEEE/ASME Trans. Mechatron. 29 (2024).

[47] Y.-L. Jin, Z.-Y. Ji, D. Zeng, X.-P. Zhang, Vwp: An efficient drl-based autonomous driving model, IEEE Trans. Multimed. 26 (2022) 2096–2108.

[48] K. Wu, H. Wang, M. Esfahani, S. Yuan, Learn to navigate autonomously through deep reinforcement learning, IEEE Trans. Ind. Electron. 69 (2021) 5342–5352.

[49] W. Zhang, Y. Zhang, N. Liu, K. Ren, P. Wang, Ipaprec: A promising tool for learning high-performance mapless navigation skills with deep reinforcement learning, IEEE/ASME Trans. Mechatron. 27 (2022) 5451–5461.

[50] J. Li, D. Isele, K. Lee, J. Park, K. Fujimura, M. Kochenderfer, Interactive autonomous navigation with internal state inference and interactivity estimation, IEEE Trans. Robot. 40 (2024) 2932–2949.