EIRM-RL: Epistemic Integrity Risk Monitoring Inspired Safe Reinforcement Learning for Trustworthy Autonomous Navigation

Yuanyuan Zhang, Yingying Wang, and Weisong Wen, *Member, IEEE*

***Abstract*—Reinforcement learning (RL) has shown great potential for autonomous navigation within internet of things (IoT) environments, where various and changing uncertainties pose significant challenges for safe, real-world deployment. Existing safe RL methods typically employ heuristic constraints while neglecting the combined impact of multiple uncertainty sources, reducing robustness and interpretability. Drawing on concepts from global navigation satellite system (GNSS) integrity monitoring, this paper proposes an epistemic integrity risk monitoring reinforcement learning (EIRM-RL) framework to enable trustworthy autonomous navigation under uncertainty. EIRM-RL extends the GNSS protection level concept to RL by utilizing an assembled world model that quantifies and incorporates sensor noise, systematic bias, and epistemic uncertainty. Furthermore, the framework continuously monitors a dynamic epistemic risk probability, which is incorporated into policy optimization as an adaptive safety constraint via Lagrangian duality. This method enables the agent to proactively avoid hazards and effectively balance safety and performance, even in highly uncertain environments. Extensive experiments demonstrate that EIRM-RL achieves superior success rates, collision avoidance, and robustness compared to state-of-the-art safe RL methods, while maintaining high efficiency.**

***Index Terms*—Reinforcement learning (RL), integrity risk monitoring, epistemic uncertainty, trustworthy autonomous navigation, unmanned ground vehicle.**

I. Introduction

With the rapid advancement of the internet of things (IoT), domains such as smart cities and autonomous transportation increasingly rely on trustworthy autonomous navigation [1]. Ensuring both high performance and safety in dynamic, uncertain, and safety-critical environments remains a central challenge for autonomous navigation systems. For instance, sensor observations are inherently vulnerable to various sources of uncertainty, including random noise [2], systematic bias [3], and environmental interference [4], which collectively pose substantial threats to the safety and reliability of autonomous agents. Consequently, developing a trustworthy navigation algorithm that balances efficiency, adaptability, and safety within multi-source uncertain environments has become a central research focus.

Model-based Trustworthy Autonomous Navigation: Model-based methods, including model predictive control (MPC) [5], [6], differential game theory [7], and adaptive robust optimization [8], form the classical optimization core of trustworthy navigation. These methods transform navigation tasks into constrained optimization problems, enabling provable safety certification of predefined safety attributes during system operation through forward reachability sets [9] or control barrier functions (CBF) [10]. However, their performance remains highly dependent on the accuracy of system dynamics and environmental modelling. Furthermore, in highly dynamic or uncertain scenarios, such as human-machine interaction or unstructured terrain, these models may fail or incur prohibitively high computational costs, limiting their real-time applicability.

Learning-based Trustworthy Autonomous Navigation: Recently, learning-based methods have garnered significant attention in autonomous navigation, particularly reinforcement learning (RL). RL enables agents to optimize navigation polices through direct environmental interaction without relying on predefined maps or manual rules, effectively handling perception-intensive complex tasks [11], [12]. Nevertheless, deploying learning-based methods, particularly RL in safety-critical scenarios, remains challenging due to the unexplainable black box nature. Uncertainties arising from sensor noise, environmental disturbances, and model limitations may distort perception and misguide policy decisions [13]. The randomness of RL exploration [14], [15] further exacerbates safety risks when observations are uncertain or degraded. More and more research has focused on safe RL methods, which are broadly categorized into safe model-free RL (SMFRL) and safe model-based RL (SMBRL) paradigms. SMFRL typically employs constrained Markov decision process (CMDP) modelling [16] and utilises Lagrangian optimization [17], [18] to enforce safety constraints. Alternative methods include distributional RL [19] and safety layers [20], which aim to confine exploration within safety boundaries. However, these methods typically rely on empirical cost estimation or expert knowledge [21], limiting their generalisability and robustness in complex environments.

On the other hand, SMBRL methods enhance sample efficiency and enable proactive risk mitigation by learning environment dynamics [22]. Trajectory prediction-based methods, such as safe model predictive control [23], [24], use learned dynamics to forecast future states and enforce safety constraints, providing theoretical guarantees under precise dynamics models. Uncertainty-aware frameworks using prob-

This work was supported in part by Hong Kong Innovation and Technology Fund-Innovation and Technology Support Program (ITF-ITSP) under the Project "Safety-Certified Multi-Source Fusion Positioning for Autonomous Vehicles in Complex Scenarios (ZPE8)", in part by Otto Poon Charitable Foundation under the project "Large Vision Model for UAV-UGV Collaborative Map Update (CDCG)", and in part by the Centre for Large AI Models (CLAIM) of the Hong Kong Polytechnic University. *(Corresponding author: Yingying Wang).*

The authors are with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: yuan-yuan.zhang@connect.polyu.hk; ying5wang@polyu.edu.hk; welson.wen@polyu.edu.hk).

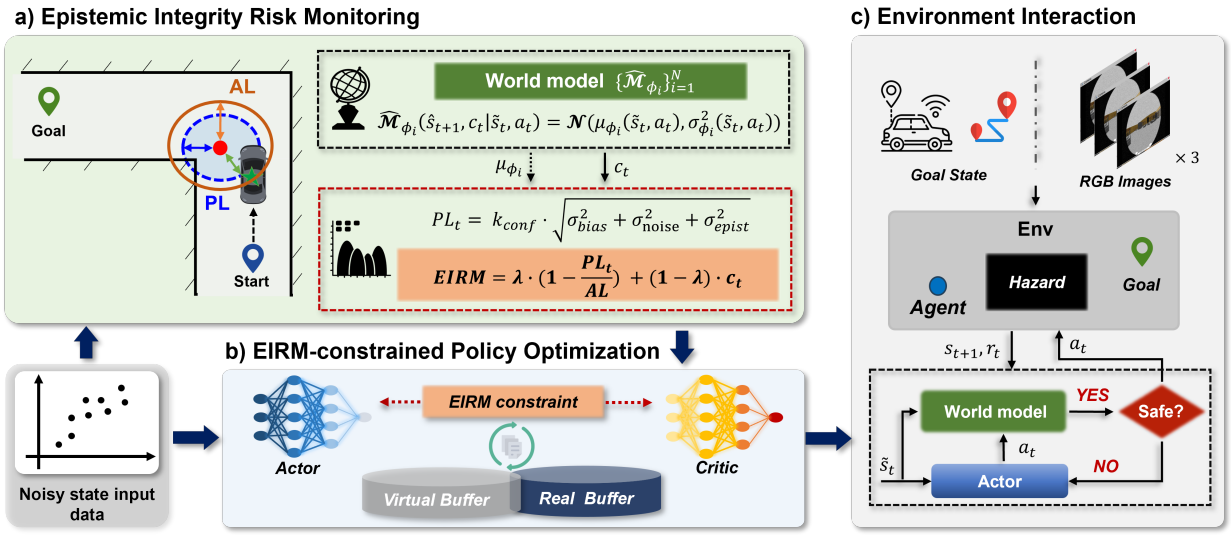


Fig. 1. Schematic diagram of the EIRM-RL: the agent receives noisy RGB images and relative goal information, extracts latent features, and interacts with the environment through an EIRM-constrained RL policy. The EIRM module quantifies multi-source uncertainty and imposes safety constraints during both policy optimization and action execution.

abilistic ensembles [25], [26] or adversarial imagination [27], further quantify epistemic uncertainty to guide safe decisions. However, most SMBRL methods handle only limited forms of uncertainty and may become overly conservative or less effective when models are inaccurate. Moreover, robust RL methods such as adversarial training [28] and ensemble-based uncertainty quantification [29] often lack principled mechanisms for real-time dynamic risk boundary control. Thus, achieving a unified and interpretable safe RL framework that systematically captures multi-source uncertainty remains an open problem.

To address this gap, we draw inspiration from global navigation satellite systems (GNSS), where integrity monitoring (IM) [30], [31] provides a rigorous framework for quantifying the reliability of state estimates and issuing timely alerts under measurement noise and systematic bias [32]. Leveraging the concepts of protection level (PL) and alert limit (AL), GNSS IM ensures reliable risk quantification and safety assurance. Inspired by this, we propose extended IM concepts to safe RL for trustworthy decision-making in dynamic and uncertain environments. In this paper, we propose a novel SMBRL framework termed **E**pistemic **I**ntegrity **R**isk **M**onitoring inspired **R**einforcement **L**earning (EIRM-RL). EIRM-RL integrates GNSS IM principles into the world model, systematically quantifying multiple risk sources through an interpretable EIRM model. This model serves as a dynamic safety constraint within policy optimization, utilizing Lagrangian duality to adaptively balance safety and performance. Our method significantly enhances safety and generalization in high-uncertain environments, providing both theoretical and practical solutions for safe RL.

The contributions of this paper are listed as follows:

(1) This paper proposes the EIRM model by integrating GNSS IM with world modelling. By extending the integrity limit PL, sensor noise, system bias, and epistemic uncertainty of the model are systematically incorporated into a unified risk assessment, enabling high-confidence uncertainty evaluation in complex dynamic environments.

(2) This paper develops a novel EIRM-RL framework that integrates EIRM-based risk monitoring into policy learning via Lagrangian dual optimization, enabling real-time adaptation between task performance and safety. This framework can adaptively adjust the safety boundary, effectively overcoming the limitations of static thresholds and empirical rules.

(3) This paper systematically evaluates the proposed EIRM-RL method in a variety of complex simulation and real-world environments. The experimental results demonstrate that EIRM-RL outperforms existing baselines in terms of safety, mission success rate, and robustness, verifying its trustworthiness and practicality in safety-critical tasks.

The remainder of this article is organized as follows. Section II provides a detailed description of the EIRM-RL framework. Following this, Section III outlines the experimental setup and evaluation metrics. The results and analysis are presented in Section IV. Finally, the main conclusions and future work are discussed in Section V.

II. Methodology

A. Overview

This section presents the EIRM-RL framework, as shown in Fig. 1. The EIRM module, built on an ensemble world model, calculates the extended PL by combining bias, noise, and model uncertainty, and compares it with an AL for real-time safety assessment. Policy optimization maximizes task rewards while keeping EIRM risk below a set threshold, ensuring only actions that meet the safety constraint are executed for robust navigation. During environment interaction, the agent encodes its state with stacked RGB images and relative goal information. To simulate worst-case sensor failures, random disturbances are added to these states, and the resulting noisy

states are used as inputs for both the world model and the policy. This design enables thorough evaluation of robustness and safety in challenging, real-world scenarios.

B. Problem Formulation

The interaction loop between safe model-based DRL and the environment can be described using a CMDP [33], represented as the 6-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, c, \gamma \rangle$. Both state space $\mathcal{S}$ and action space $\mathcal{A}$ are bounded and continuous; the initial state set is denoted as $\mathcal{S}_0$; $\mathcal{P}$ represents the state transition probability; $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward; c is the cost function defined as $c = 1$ if the state constraint $h(\mathbf{s}_t) \leq 0$ is violated and 0 otherwise; $\gamma \in (0, 1)$ serves as the discount factor; a deterministic policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ chooses action variable $\mathbf{a}_t$ at state variable $\mathbf{s}_t$ at time step t. The CMDP computes the cumulative reward by constructing an action-value function $Q^\pi(\mathbf{s}_t, \mathbf{a}_t)$ which can be expressed using the Bellman equation [34]:

$$\begin{aligned} Q^\pi(\mathbf{s}_t, \mathbf{a}_t) &= r(\mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}\left[\gamma \max_{\mathbf{a}_t} \mathbb{E}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})]\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]. \end{aligned} \tag{1}$$

1) State space: The state space $\mathcal{S}$ consists of two components: the visual state $\mathcal{S}_{\text{vis}}$, which represents information about the surrounding environment, and the relative information $\mathcal{S}^{\Delta}_{\text{goal}}$ between the unmanned ground vehicle (UGV) agent and the goal point. This can be represented as

$$\mathcal{S} = [\mathcal{S}_{\text{vis}}, \mathcal{S}^{\Delta}_{\text{goal}}]. \tag{2}$$

Specifically, $\mathcal{S}_{\text{vis}}$ is captured by stacking the most recent three raw RGB images $\mathbf{i}$ captured by the fisheye camera

$$\mathbf{I}_t = [\mathbf{i}_{t-2}, \mathbf{i}_{t-1}, \mathbf{i}_t] \in \mathbb{R}^{3\times H\times W\times 3}, \tag{3}$$

where $\mathbf{I}_t$ denotes the stacked RGB image at time t, with a height H of 128 pixels and a width W of 160 pixels. This image sequence is processed by a deep convolutional encoder $\Phi_{\phi_{\text{conv}}}$ to extract a latent representation

$$\mathbf{z}_t = \Phi_{\phi_{\text{conv}}}(\mathbf{I}_t), \tag{4}$$

where $\mathbf{z}_t \in \mathbb{R}^L$ is an L-dimensional visual feature vector and ϕ_{conv} denotes the encoder parameters. Thus,

$$\mathcal{S}_{\text{vis}} = [\mathbf{z}_t]. \tag{5}$$

In addition, we add pixel-level Gaussian noise to the input images to simulate visual disturbances in real-world environments, thereby learning a more robust and transferable navigation strategy.

The goal-related state $\mathcal{S}^{\Delta}_{\text{goal}}$ is represented by the relative features between the UGV agent's real-time state and the goal point's state

$$\mathcal{S}^{\Delta}_{\text{goal}} = [\Delta d_t, \Delta\varphi_t], \tag{6}$$

where Δd_t and $\Delta\varphi_t$ represent the normalized relative distance and heading error, respectively. The normalized relative distance Δd_t is calculated by

$$\Delta d_t = \frac{\|\mathbf{p}_{\text{goal}} - \mathbf{p}_t\|_2}{d_{\max}}, \tag{7}$$

where $\mathbf{p}_{\text{goal}} = (p^x_{\text{goal}}, p^y_{\text{goal}})$ and $\mathbf{p}_t = (p^x_t, p^y_t)$ are the 2D coordinates of the goal point and the UGV at time t, $\|\cdot\|_2$ represent euclidean norm operation, and $d_{\max}$ denotes the normalized maximum distance gap. The relative heading $\Delta\varphi_t$ is obtained by computing the angular difference between the UGV's orientation and the direction towards the goal. The normalized relative heading error is calculated as

$$\Delta\varphi_t = \frac{1}{\pi} \cdot \text{wrap}\left(\arctan 2(p^y_{\text{goal}} - p^y_t,\, p^x_{\text{goal}} - p^x_t) - \theta_t\right), \tag{8}$$

where θ_t is the UGV's heading angle, and $\text{wrap}(\cdot)$ wraps the angle to $(-\pi, \pi]$.

2) Action space: The action space $\mathcal{A}$ is defined as the control variables of the UGV motion, including the linear velocity and the angular velocity

$$\mathcal{A} = [\mathbf{v}_t, \boldsymbol{\omega}_t], \tag{9}$$

where $\mathbf{v}_t \in [0, 1.2]$m/s denotes the linear velocity, and $\boldsymbol{\omega}_t \in [-2, 2]$rad/s denotes the angular velocity. The bounds for these commands are determined based on the specific characteristics of the experimental platform, specifically the Agilex ScoutMini mobile robot (see Section III-A2). These two actions are used to achieve continuous control of the UGV. The actions are sent to the environment every 0.1 seconds.

3) Reward function: The reward function defines the agent's learning objective, balancing task efficiency and collision avoidance through continuous guidance signals and event-driven incentives.

First, the proximity reward r_{prox} will encourage the agent to reduce the Euclidean distance to the goal:

$$r_{\text{prox}} = \alpha_p(\|\mathbf{p}_{t-1} - \mathbf{p}_{\text{goal}}\|_2 - \|\mathbf{p}_t - \mathbf{p}_{\text{goal}}\|_2), \tag{10}$$

where α_p is a scaling factor.

Second, the UGV agent will receive a motion efficiency reward r_{ctrl} that promotes smooth navigation by rewarding forward velocity $\mathbf{v}_t$ and penalizing angular velocity $\boldsymbol{\omega}_t$:

$$r_{\text{ctrl}} = \mathbf{v}_t - \alpha_c|\boldsymbol{\omega}_t|, \tag{11}$$

where α_c controls the penalty for rotation.

Third, a goal-reaching reward r_{succ} will be given when the UGV reaches the goal point within a distance threshold. Similarly, a collision penalty r_{coll} imposes a negative reward when a collision is detected. They are designed as follows:

$$r_{\text{succ}} = \begin{cases} C_1, & \text{if } \Delta d_t \leq \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

$$r_{\text{coll}} = \begin{cases} C_2, & \text{if collision occurs,} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

where ϵ is a predefined distance threshold, C_1 (e.g., 100) and C_2 (e.g., -100) are constant value.

In summary, at each time step t, the reward r can be formulated as

$$r = r_{\text{prox}} + r_{\text{ctrl}} + r_{\text{succ}} + r_{\text{coll}}. \tag{14}$$

C. Epistemic Integrity Risk Monitoring Algorithm

In real-world applications, observed states are frequently affected by noise, including random errors, sensor bias, and drift. To emulate such sensor imperfections and potential failures, random Gaussian noise with varying bias is added to the input state $\mathbf{s}_t$ of the world model during training. The noisy observed state $\widetilde{\mathbf{s}}_t$ is modeled as

$$\widetilde{\mathbf{s}}_t = \mathbf{s}_t + \boldsymbol{b}_t + \epsilon_t, \tag{15}$$

where $\mathbf{s}_t$ denotes the true underlying state, $\boldsymbol{b}_t$ represents a bounded additional bias sampled from a uniform distribution $\boldsymbol{b}_t \sim \mathcal{U}(\boldsymbol{b}_t^{\min}, \boldsymbol{b}_t^{\max})$, and $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2)$ is modeled as zero-mean Gaussian noise with a standard deviation σ_t drawn from a uniform range $\sigma_t \sim \mathcal{U}(\sigma_t^{\min}, \sigma_t^{\max})$. The bias $\boldsymbol{b}_t$ reflects systematic errors commonly found in sensor measurements, such as those caused by landmark detection inaccuracies, global positioning system (GPS) signal reflections, or drift in inertial measurement units.

To capture both epistemic and aleatoric uncertainty in environmental dynamics and safety cost prediction, EIRM-RL adopts an ensemble of N diagonal Gaussian models parameterized by ϕ_i to model the environment dynamics, denoted as $\{\hat{\mathcal{M}}_{\phi_i}\}_{i=1}^{N}$. Each model $\hat{\mathcal{M}}_{\phi_i}$ predicts the next state $\hat{\mathbf{s}}_{t+1}$ and the immediate safety cost c_t as a joint Gaussian distribution conditioned on the current state $\widetilde{\mathbf{s}}_t$ and action $\mathbf{a}_t$:

$$\hat{\mathcal{M}}_{\phi_i}(\hat{\mathbf{s}}_{t+1}, c_t | \widetilde{\mathbf{s}}_t, \mathbf{a}_t) = \mathcal{N}(\mu_{\phi_i}(\widetilde{\mathbf{s}}_t, \mathbf{a}_t), \sigma_{\phi_i}^2(\widetilde{\mathbf{s}}_t, \mathbf{a}_t)), \tag{16}$$

where c_t denotes the probability of safety violations, $\mu_{\phi_i}(\widetilde{\mathbf{s}}_t, \mathbf{a}_t)$ and $\sigma_{\phi_i}^2(\widetilde{\mathbf{s}}_t, \mathbf{a}_t)$ denote the predicted mean and variance matrix of the predicted Gaussian distribution, respectively.

During world model training, each sub-model in the ensemble is initialized randomly and updated with different mini-batches selected from the real environment interaction reply buffer $\mathcal{B}_{\text{real}}$. The models are optimized by minimizing the negative log-likelihood loss:

$$\mathcal{L}_{\hat{\mathcal{M}}}(\phi_i) = -\mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t, c_t, \mathbf{s}_{t+1}) \sim \mathcal{B}_{\text{real}}} \left[\log \hat{\mathcal{M}}_{\phi_i}(\hat{\mathbf{s}}_{t+1}, c_t \mid \widetilde{\mathbf{s}}_t, \mathbf{a}_t) \right], \tag{17}$$

where $\mathbf{s}_{t+1}$ represents the true next state. Therefore, the world model generates synthetic transitions via recursive sampling:

$$\hat{\mathbf{s}}_{t+1}, c_t \sim \hat{\mathcal{M}}_{\phi_i}(\cdot | \widetilde{\mathbf{s}}_t, \mathbf{a}_t). \tag{18}$$

The synthetic transition is injected into the virtual replay buffer $\mathcal{B}_{\text{virt}}$ and used together with the real transition for policy training to reduce the reliance on real environment interaction and reduce the risk of unsafe exploration. By enhancing the training data, the synthetic trajectory improves the sample efficiency and provides uncertainty quantification support for subsequent risk monitoring and decision-making.

Additionally, inspired by the IM concept of GNSS, EIRM-RL proposes the EIRM model, which integrates the IM framework with the world model to support safety-critical decision-making. In GNSS, system integrity is defined as the degree of trust in state estimates, ensuring the system promptly issues alerts when the information is unreliable or degraded [32]. The IM framework is based on two core concepts: PL and AL [35]. IM translates observation or model uncertainty into a high-confidence bound, PL, which represents the maximum potential deviation between the predicted and true states based on error propagation theory. By comparing PL to AL, an interpretable safety standard is established: PL $\leq$ AL. This mechanism quantifies the probability of undetected unsafe states as [31]:

$$P_{\text{IR}} = P(|\hat{\mathbf{s}} - \mathbf{s}| > AL \wedge \text{No Alert}) < \delta, \tag{19}$$

where P_{IR} denotes the probability of an undetected integrity risk, $\hat{\mathbf{s}}$ and $\mathbf{s}$ are the estimated state and the true state, respectively, δ is specific threshold.

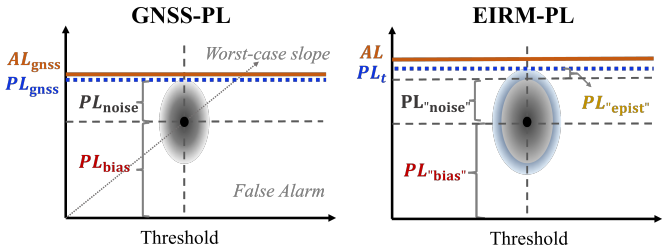


Fig. 2. Visual comparison of GNSS-PL and EIRM extended-PL. **Left:** GNSS PL_{gnss} (dashed blue line) reflects the uncertainty range of position information estimation (black ellipse) by accumulating PL_{noise} and PL_{bias}, which is lower than AL_{gnss} (solid orange line). The threshold is defined by the worst-case slope to avoid a false alarm. **Right:** EIRM PL_{t} (dashed blue line) inspired by PL_{gnss} incorporates $PL_{\text{"bias"}}$, $PL_{\text{"noise"}}$ and $PL_{\text{"epist"}}$.

The conventional PL for GNSS is based on statistical constraints on position estimation uncertainty, usually characterized by a covariance matrix generated from measurement errors, and is used to protect against worst-case position errors due to measurement faults and noise. However, as shown in Fig. 2, the PL_{gnss} of GNSS focuses only on measurement noise PL_{noise} and systematic bias PL_{bias}, whereas the extended PL_t of EIRM-RL goes further by integrating model epistemic uncertainty $PL_{\text{"epist"}}$, which can cover more complex decision scenario, where uncertainty may come from other sources such as model predictions, sensor biases, and complex environmental dynamics. Specifically, the extended PL is defined as the root-sum-square (RSS) of these uncertainties, scaled by :

$$PL_t = k_{\text{conf}} \cdot \sqrt{\sigma_{\text{bias}}^2 + \sigma_{\text{noise}}^2 + \sigma_{\text{epist}}^2}, \tag{20}$$

where σ_{bias}^2, σ_{noise}^2, and σ_{epist}^2 denote the variances attributed to systematic bias, sensor noise, and model epistemic uncertainty, respectively. The scaling factor $k_{\text{conf}} = Q^{-1}(1 - \frac{\eta}{2})$ corresponds to the quantile of the standard normal distribution that determines the desired statistical confidence level (e.g., $k_{\text{conf}} = 1.96$ when $\eta = 0.05$, corresponding to 95% confidence), with $Q^{-1}(\cdot)$ being the inverse cumulative distribution function of the standard normal distribution. This extended PL significantly improves the coverage of multi-source risks and is particularly suitable for epistemic environments such as RL.

First, the systematic bias uncertainty is used to capture the long-term accumulated bias risk, such as sensor drift and calibration error. It is estimated by the upper bound of the

uniform distribution variance of the known bias intervals $\boldsymbol{b}_t^{\min}$ and $\boldsymbol{b}_t^{\max}$:

$$\sigma_{\text{bias}}^2 = \frac{1}{12}\left\|\boldsymbol{b}_t^{\max} - \boldsymbol{b}_t^{\min}\right\|^2. \tag{21}$$

This explicit characterization ensures that worst-case bias effects are reflected in the final risk estimate.

Second, the noise uncertainty arises from the irreducible randomness inherent to sensor measurements and is quantified by averaging the predictive covariance matrices across the model ensemble.

$$\sigma_{\text{noise}}^2 = \frac{1}{N}\sum_{i=1}^{N}\sigma_{\phi_i}^2(\widetilde{\mathbf{s}}_t, \mathbf{a}_t). \tag{22}$$

Finally, the epistemic uncertainty reflects the prediction divergence caused by the limited generalization ability of the model or the limited distribution of training samples. To this end, we first calculate the ensemble mean $\bar{\boldsymbol{\mu}}_t$ of the output means of all sub-models under the input $(\widetilde{\mathbf{s}}_t, \mathbf{a}_t)$:

$$\bar{\boldsymbol{\mu}}_t = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\mu}_{\phi_i}(\widetilde{\mathbf{s}}_t, \mathbf{a}_t), \tag{23}$$

where $\boldsymbol{\mu}_{\phi_i}$ denotes the predicted mean from the i-th model. Based on this, epistemic uncertainty can be quantitatively expressed by the variance of the Euclidean distance between the mean of each sub-model and the ensemble mean:

$$\sigma_{\text{epist}}^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{\mu}_{\phi_i}(\widetilde{\mathbf{s}}_t, \mathbf{a}_t) - \bar{\boldsymbol{\mu}}_t\right\|_2, \tag{24}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, which measures the degree of dispersion between model outputs.

The EIRM model further incorporates the normalized immediate safety cost

$$\text{EIRM}(\mathbf{s}_t, \mathbf{a}_t) = \lambda \cdot (1 - \frac{PL_t}{AL}) + (1-\lambda)\cdot c_t, \tag{25}$$

where $\lambda \in [0,1]$ balances uncertainty-based risk and empirical safety cost and $c_t \in [0,1]$. The AL defines the maximum allowable deviation between the estimated and true states before triggering a safety alert, serving as a quantitative boundary for integrity assurance. In this work, AL is determined according to the UGV's geometric clearance and task-specific safety requirements. The specific AL values for different environments are set out in Table III.

D. EIRM-Constrained Policy Optimization

The policy optimization goal of EIRM-RL is to balance task performance and safety, maximizing the cumulative reward while satisfying the constraints on EIRM risk during navigation. Specifically, policy optimization is defined as the following dual constrained optimization problem

$$\max_{\pi}\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right] \quad \text{s.t.} \quad \mathbb{E}\left[\sum_{t=0}^{\infty}\text{EIRM}(\mathbf{s}_t, \mathbf{a}_t)\right] \le \epsilon_{\text{EIRM}}, \tag{26}$$

where ϵ_{EIRM} denotes a predefined safety threshold.

To solve this problem efficiently, EIRM-RL uses the Lagrangian dual method to transform it into an unconstrained optimization problem. Using Lagrange duality, the constrained optimization is reformulated as

$$\mathcal{L}(\pi, \beta) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(\mathbf{s}_t, \mathbf{a}_t) + \beta\left(\epsilon_{\text{EIRM}} - \text{EIRM}(\mathbf{s}_t, \mathbf{a}_t)\right)\right)\right], \tag{27}$$

where $\beta \ge 0$ is the Lagrange multiplier that dynamically adjusts the trade-off between reward maximization and safety constraints. The dual optimization problem is expressed as

$$\min_{\beta\ge 0}\max_{\pi}\mathcal{L}(\pi, \beta). \tag{28}$$

The optimization process alternates between updating the policy, evaluating the value function, and adjusting the Lagrange multiplier, ensuring a balance between task performance and safety.

During training, β is updated iteratively to ensure that the safety constraint is satisfied while maximizing the cumulative reward. The dual update rule is expressed as:

$$\beta \leftarrow \max\left(0, \beta + l_d\left(\mathbb{E}\left[\sum_{t=0}^{\infty}\text{EIRM}(\mathbf{s}_t, \mathbf{a}_t)\right] - \epsilon_{\text{EIRM}}\right)\right), \tag{29}$$

where $l_d > 0$ is the dual learning rate for β. This update ensures that β increases when the cumulative EIRM violations exceed the safety threshold ϵ_{EIRM}, thereby enforcing the safety constraint, and decreases otherwise to avoid overly conservative behaviors.

The computational procedure for the proposed EIRM-RL algorithm is detailed in Algorithm 1.

III. Experimental Validation

To evaluate the effectiveness and practicality of the proposed EIRM-RL framework, we designed a goal-driven mapless autonomous navigation task in both simulated and real environments. The goal is to guide a ground robot to a specific target location while avoiding collisions under various uncertain conditions.

A. Experimental Platform and Scenarios

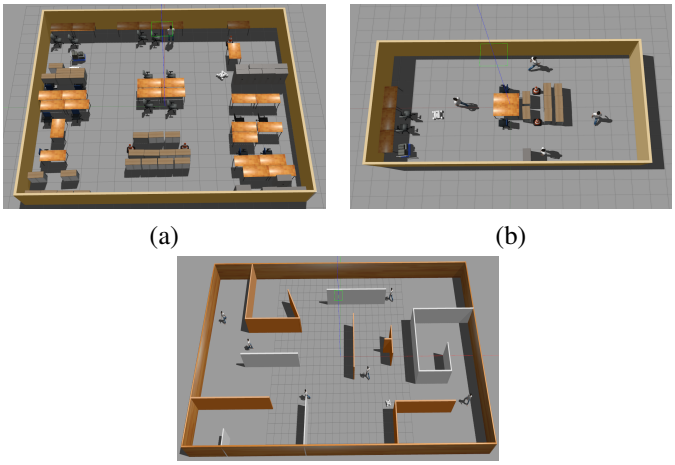
1) Simulation Experiment: Simulation experiments are conducted in Gazebo using a differential-drive UGV model, as illustrated in Fig. 3. Three simulated environments are designed to evaluate the proposed framework under increasing complexity. The first is a static office scenario with cluttered obstacles such as desks and chairs, as shown in Fig. 3(a). The second is a dynamic canteen scenario containing both static objects and multiple pedestrians moving at constant velocities uniformly sampled between 0.5 m/s and 1.5 m/s, as shown in Fig. 3(b). At the beginning of each episode, the UGV's start and goal positions are randomly assigned to enhance scene diversity. The UGV is equipped with a fisheye RGB camera for perception and a simulated laser rangefinder for collision detection. Sensor uncertainties are introduced by adding Gaussian noise to camera images and state observations, with noise parameters varied to assess robustness, as detailed in Table I. Level-0 employs zero-mean Gaussian noise, while Level-1 and

Algorithm 1 EIRM-RL algorithm

Input: Real buffer $\mathcal{B}_{\text{real}}$; Virtual buffer $\mathcal{B}_{\text{virt}}$; Ensemble size N; Confidence level α; AL; EIRM safety threshold ϵ_{EIRM}; Risk weight λ

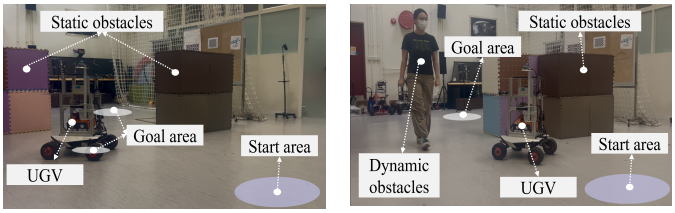
Output: Optimized policy π

Initialize world model ensemble $\{\hat{\mathcal{M}}_{\phi_i}\}_{i=1}^{N}$ with random parameters ϕ_i, policy π_{init} with parameters ϕ_π, Lagrange multiplier $\beta \leftarrow 1$, replay buffers $\mathcal{B}_{\text{real}}$ and $\mathcal{B}_{\text{virt}}$
for each training iteration **do**
  Current state $\mathbf{s}_t$ from the environment
  Sample bias $\boldsymbol{b}_t \sim \mathcal{U}(\boldsymbol{b}_t^{\min}, \boldsymbol{b}_t^{\max})$ and noise $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2)$
  Compute noisy state $\widetilde{\mathbf{s}}_t = \mathbf{s}_t + \boldsymbol{b}_t + \boldsymbol{\epsilon}_t$ ▷ Eq. (12)
  Execute action $\mathbf{a}_t = \pi(\widetilde{\mathbf{s}}_t)$ in environment
  Observe reward r_t, next state $\mathbf{s}_{t+1}$
  Store transition $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ in $\mathcal{B}_{\text{real}}$
  for $i = 1$ to N **do**
    Sample mini-batches from $\mathcal{B}_{\text{real}}$
    Update ϕ_i by minimizing negative log-likelihood loss ▷ Eq. (15)
  end for
  Sample $(\widetilde{\mathbf{s}}_t, \mathbf{a}_t)$ from buffer and Randomly select $i \in \{1, \ldots, N\}$
  Sample $(\hat{\mathbf{s}}_{t+1}, c_t) \sim \hat{\mathcal{M}}_{\phi_i}(\cdot|\widetilde{\mathbf{s}}_t, \mathbf{a}_t)$ ▷ Eq. (16)
  Store $(\widetilde{\mathbf{s}}_t, \mathbf{a}_t, c_t, \hat{\mathbf{s}}_{t+1})$ in $\mathcal{B}_{\text{virt}}$
  Compute σ^2_{bias}, σ^2_{noise} and σ^2_{epist} ▷ Eq. (20), (21), (23)
  Compute $k_{\text{conf}} = \Phi^{-1}(1 - \frac{\eta}{2})$
  Compute PL_t and EIRM$(\mathbf{s}_t, \mathbf{a}_t)$ ▷ Eq. (19), (24)
  Update policy parameters ϕ_π ▷ Eq. (27)
  Update Lagrange multiplier β ▷ Eq. (29)
end for
return π



(a) (b) (c)

Fig. 3. Illustration of the simulation environments: (a) Static office scenario. (b) Dynamic canteen scenario. (c) Unseen maze scenario for generalization test.



(a) (b)

Fig. 4. Illustration of the experiment scenarios: (a) Real-world static environment with foam obstacles and two goal points. (b) Real-world dynamic environment with human participants. The start area, goal areas, UGV and obstacles are marked in each scenario.

Level-2 utilize biased Gaussian perturbations with increasing severity, with Level-2 indicating a state close to sensor failure.

In addition, an unseen maze-like environment, shown in Fig. 3(c), is designed to test policy generalization in unseen spatial configurations. The scenario features irregular corridors, multiple blind corners, and randomly placed dynamic obstacles, creating a highly unstructured and previously unseen navigation environment. This scenario contains irregular corridor layouts with passage widths of 0.6-1.2 meters and dynamic obstacles that require both local avoidance and long-range planning. The UGV is initialized randomly near the entrance, while the goal is located at the far end, emphasizing path efficiency and adaptability without applying any domain randomization or fine-tuning.

The system is integrated using the robot operating system (ROS), which also provides odometry-based goal information. All simulations run on a workstation equipped with an Intel Core i7-13700K CPU, 32 GB RAM, and an NVIDIA RTX 4070 GPU.

2) Sim-to-Real Experiment: The real-world platform is based on the Agilex ScoutMini robot, as illustrated in Fig. 4(c)-(d). The UGV is equipped with a fisheye RGB camera (128×160 resolution, 210° field of view), an IMU, and a NVIDIA Jetson Orin module for real-time inference, powered by a mobile battery. ROS manages system integration and data flow, while low-level control is performed via CAN communication. In the static scene (Fig. 4(c)), foam blocks are arranged to create narrow passages, and the UGV is required to complete a two-stage navigation task. Specifically, it starts from the origin, first navigates to the intermediate goal at $(-3, 3)$, pauses for two seconds, and then continues to the final goal at $(-0.5, 6)$. Navigation is considered successful if the UGV reaches within 0.5 meters of the specified goal point. In the dynamic environment (Fig. 4(d)), human participants move freely and randomly within the experimental area, introducing unpredictable disturbances and dynamic obstacles. It should be mentioned that the UGV's ego-pose is estimated using the IMU in combination with wheel odometry for dead-reckoning localization. The IMU data provides a continuous estimate of the vehicle's linear acceleration and angular velocity, which is integrated for pose estimation in real time. Over the longest experimental trajectory of approximately 8 meters, the absolute trajectory error (ATE) exhibits a root mean square error (RMSE) of about 0.05 meters, indicating that the accumulated drift remained. This positioning accuracy ensures practical feasibility in real-world reasoning applications. These experimental setups enable a comprehensive evaluation of EIRM-RL's safety, robustness, and generalization in both controlled and dynamic real-world settings.

TABLE I
NOISE PARAMETERS

Noise Type	Parameter	Bias (μ)	Std (σ)
Training (Normal*)	Position (m)	0	0.05
	Linear Vel. (m/s)	0	0.05
	Angular Vel. (rad/s)	0	0.05
Training (High**)	Position (m)	[0, 2]	[0.05, 0.5]
	Linear Vel. (m/s)	[0, 1]	[0.05, 0.5]
	Angular Vel. (rad/s)	[0, 0.5]	[0.05, 0.1]
Test (Level-1)	Position (m)	1.5	0.2
	Linear Vel. (m/s)	0.5	0.1
	Angular Vel. (rad/s)	0.2	0.07
Test (Level-2)	Position (m)	2	0.5
	Linear Vel. (m/s)	1	0.5
	Angular Vel. (rad/s)	0.5	0.1

* Normal: Simulate slight GPS or odometry errors. Test Level-0=Normal.
** High: Simulate more challenging sensor interference.

B. Baseline Methods

All baseline methods are implemented within the actor-critic framework and use the same input modalities and action spaces for fair comparison. Moreover, all baselines employ the same visual encoder architecture and share identical reward functions.

1) *SAC [36]*: A widely adopted off-policy RL algorithm for continuous control. We implement a goal-conditioned soft actor-critic agent with a convolutional scene encoder, where goal information is embedded into the latent representation.
2) *SAC-Lag [37]*: An extension of SAC that incorporates Lagrangian-based constrained optimization. It simultaneously optimizes reward and cost value functions, updating the Lagrange multiplier via dual ascent.
3) *MBPO [38]*: A model-based policy optimization algorithm that updates the policy using short rollouts generated by a learned model, without explicit safety constraints.
4) *SMBPO [27]*: A safety-aware variant of MBPO. It is noteworthy that, unlike RIRM-RL, SMBPO incorporates only the world model uncertainty σ^2_{epist} and predicted safety costs $c_t \in [0, 1]$ into risk constraints. Policy optimization within SMBPO is conducted via the Lagrange method.
5) *MPC-CBF [10]*: A representative model-based control baseline combining MPC with CBF theory. The MPC module generates locally optimal trajectories by minimizing a cost function that balances control effort and goal-tracking error, while the CBF enforces formal safety guarantees by constraining system evolution within safe states. Specifically, the safety constraint is formulated as $\dot{h}(s) + \mu h(s) \geq 0$, where $h(s)$ defines the safety boundary and $\mu \in (0, 1)$ regulates the admissible convergence rate toward it. This baseline is therefore included to benchmark EIRM-RL against a deterministic, optimization-driven paradigm that represents the state-of-the-art in model-based safe control.

C. Metrics

1) Training metrics: We assess the comprehensive training performance of the policy with three metrics: (1) **mean reward** reflects the overall performance of various intermediate agents during an episode; (2) **success rate (SR)** indicates the proportion of episodes in which the agent reaches the goal point without any collisions within the maximum allowed timesteps; and (3) **collision rate (CR)** shows the proportion of episodes in which the agent collides with obstacles or other agents before reaching the goal.

2) Test metrics: After training, we use five metrics to test the post-trained RL agent: (1) **SR**; (2) **CR**; (3) **average velocity (AV)** measures the mean velocity during task execution, reflecting the efficiency of movement; (4) **max velocity (MV)** records the max velocity reached by the agent in an episode, indicating the policy's aggressiveness or safety margin; (5) **computing time (CT)** reflects the computational load of different algorithms. All test metrics are calculated based on 100 evaluation episodes.

D. Neural Network and Hyperparameter Configuration

We utilize the SAC architecture as our EIRM-RL agent. The structural details of the RL neural network employed in the algorithm are outlined in Table II. The hyperparameters utilized in the EIRM-RL algorithm are described in detail in Table III.

TABLE II
NEURAL NETWORK STRUCTURE AND PARAMETERS

Parameters	Value
Network architecture	5 layers (3 conv + 2 FC)
Input image shape	[160,128,3]
Convolution filter features	[32,64,128] (kernel size 5×5, stride=2, padding=2)
Conv output dimensions	[20,16,128]
Flattened features	[40,962]
Fully connected layer	[512,256,2]
Output activation	Sigmoid + Tanh
Activation function	Leaky ReLU
Implementation	PyTorch

TABLE III
HYPERPARAMETERS OF EIRM-RL ALGORITHM

Parameters	Value
Replay buffer size ($\mathcal{B}_{\text{real}}$, $\mathcal{B}_{\text{virt}}$)	2e4
Actor learning rate (l_a)	0.001
Critic learning rate (l_c)	0.001
Dual learning rate (l_d)	0.0001
World model learning rate (l_w)	0.001
Discount factor (γ)	0.999
Soft goal update coefficient (τ)	0.005
Reward term weight (α_p, α_c, C_1, C_2)	20, 2, 100, -100
EIRM model's weight (λ)	0.2
Safety threshold (ϵ_{EIRM})	0.4
Replay buffer size ($\mathcal{B}_{\text{real}}$, $\mathcal{B}_{\text{virt}}$)	2e4
Actor-critic batch size	32
World model batch size	64
Confidence factor (k_{conf})	1.96 (95% confidence)
Alert limit (AL)	0.5 (static), 0.3 (dynamic)

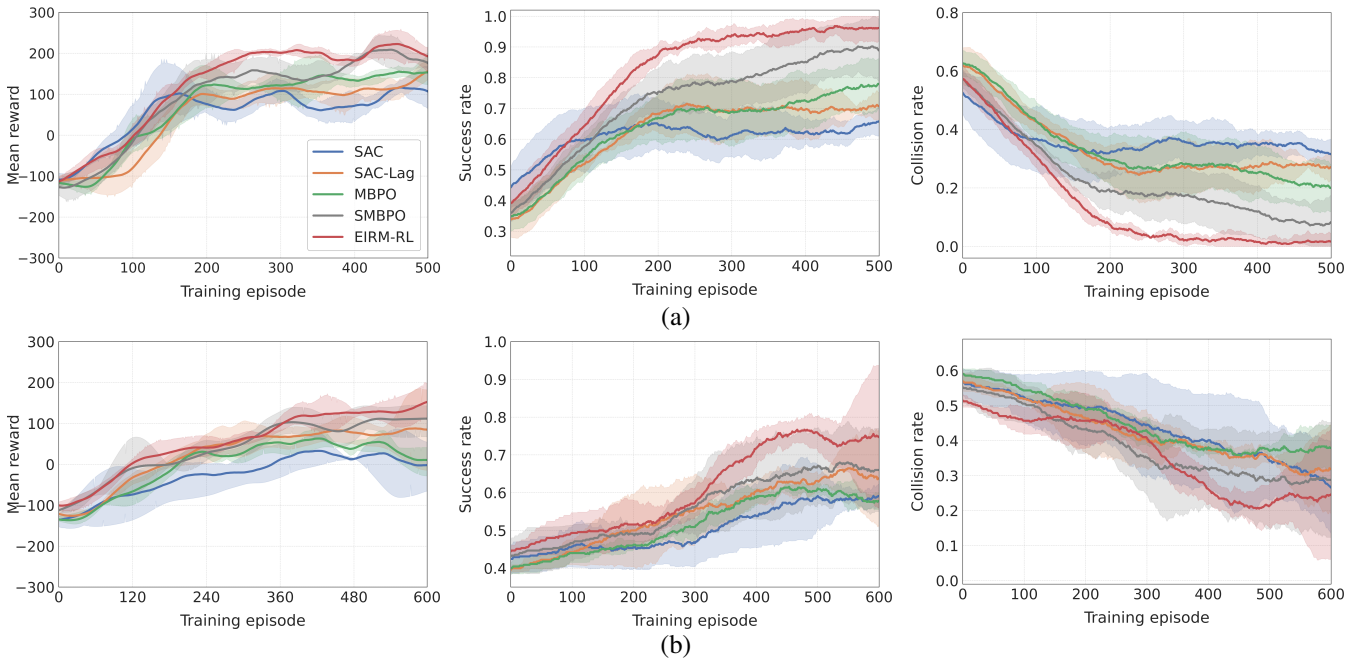


Fig. 5. Comprehensive training curves comparison in static and dynamic environments. (a) Shows the mean reward (left), success rate (middle), and collision rate (right) in a static office environment. (b) Shows the corresponding metrics in a dynamic environment.

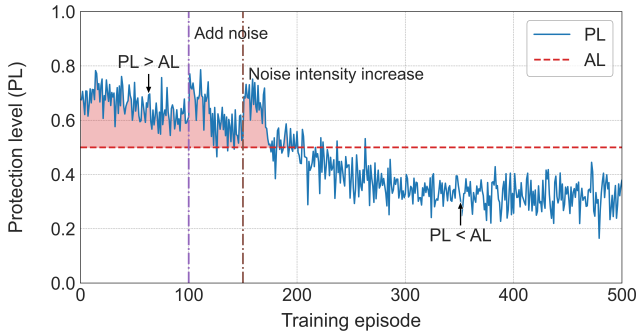


Fig. 6. Evolution of PL and AL during the EIRM-RL training process in static environment. The PL curve illustrates the progressive reduction of uncertainty as the policy converges, while its comparison with the fixed AL boundary demonstrates compliance with the safety constraint.

IV. Results and Analysis

A. Simulation Experiments

1) Training evaluation: Fig. 5 shows the training performance of the EIRM-RL algorithm and mainstream baseline algorithms in static and dynamic environments. The solid line in the figure represents the mean of each indicator, and the shaded part represents the standard deviation.

In the static environment, as shown in Fig. 5(a), the average reward of EIRM-RL is improved by approximately 38.89%, 47.06%, 31.58%, and 13.64% compared with the four baseline algorithms SAC, SAC-Lag, MBPO, and SMBPO, respectively; the final success rate is increased by 18.75%, 23.38%, 11.76%, and 5.56%, respectively; and the collision rate is reduced by approximately 72.22%, 75.00%, 50.00%, and 37.50%, respectively.

In a more challenging dynamic environment, as shown in Fig. 5(b), compared with SAC, SAC-Lag, MBPO, and SMBPO, EIRM-RL's average reward increased by about 60.00%, 50.01%, 20.00%, and 11.63%, respectively; the final success rate increased by about 32.86%, 27.40%, 12.05%, and 3.33%, respectively; and the collision rate decreased by about 63.16%, 65.00%, 41.67%, and 22.22%, respectively. Especially in terms of collision rate, EIRM-RL has a significant safety advantage, effectively ensuring the robustness and safety of the agent in complex environments.

To further visualize and explain the internal learning dynamics of EIRM-RL, Fig. 6 illustrates the temporal evolution of the PL and AL within a static environment during policy training. At the early training stage (before approximately episode 150), the PL value frequently exceeds the AL threshold due to the immature policy and high model uncertainty. As training proceeds, the PL curve steadily declines and finally converges below the fixed AL boundary, clearly demonstrating the gradual reduction in uncertainty and the establishment of a safety-compliant policy. The highlighted noise injection interval in the figure further validates the EIRM module's adaptability under increased observation disturbances, where PL temporarily rises but rapidly re-stabilizes, confirming the model's robustness and real-time integrity monitoring capability.

These visual results illustrate how the safety constraint are internalized during policy optimization. The narrowing PL-AL gap indicates policy convergence and showcases the EIRM mechanism's dynamic calibration of epistemic and stochastic uncertainties. Together with the metrics in Fig. 5(a)-(b), this visualization demonstrates the synchronized improvement of safety assurance and reward maximization throughout training.

TABLE IV
COMPARISON OF TEST STATISTICAL RESULTS IN TWO SCENARIOS, INCLUDING THE MEAN VALUE AND STANDARD DEVIATION (IN BRACKETS).

Methods	Metric	Scenario (a)			Scenario (b)		
		Level-0	Level-1	Level-2	Level-0	Level-1	Level-2
SAC	SR	0.70 (0.01)	0.63 (0.01)	0.60 (0.02)	0.61 (0.05)	0.58 (0.08)	0.57 (0.12)
	CR	0.27 (0.01)	0.32 (0.01)	0.33 (0.02)	0.27 (0.05)	0.31 (0.08)	0.33 (0.12)
	AV	1.10 (0.02)	1.08 (0.04)	1.06 (0.07)	**1.00 (0.05)**	**1.09 (0.03)**	**1.08 (0.04)**
	MV	**1.22 (0.01)**	**1.20 (0.02)**	**1.20 (0.02)**	1.21 (0.06)	1.20 (0.07)	1.15 (0.16)
	CT	**0.48 (0.05)**	**0.49 (0.11)**	**0.52 (0.08)**	0.67 (0.07)	0.68 (0.09)	0.70 (0.03)
SAC-Lag	SR	0.72 (0.02)	0.67 (0.01)	0.64 (0.03)	0.69 (0.04)	0.64 (0.05)	0.60 (0.05)
	CR	0.25 (0.02)	0.28 (0.01)	0.31 (0.03)	0.30 (0.04)	0.32 (0.05)	0.38 (0.05)
	AV	**1.12 (0.10)**	**1.11 (0.12)**	**1.09 (0.13)**	0.89 (0.18)	0.87 (0.20)	0.83 (0.25)
	MV	1.20 (0.02)	1.19 (0.04)	1.19 (0.00)	1.22 (0.11)	1.18 (0.09)	1.16 (0.13)
	CT	0.53 (0.00)	0.54 (0.01)	0.54 (0.01)	**0.57 (0.03)**	**0.59 (0.03)**	**0.60 (0.04)**
MBPO	SR	0.86 (0.03)	0.78 (0.03)	0.69 (0.04)	0.63 (0.02)	0.57 (0.03)	0.56 (0.03)
	CR	0.11 (0.03)	0.23 (0.03)	0.29 (0.04)	0.32 (0.02)	0.37 (0.03)	0.41 (0.03)
	AV	1.08 (0.05)	1.07 (0.04)	1.05 (0.06)	0.90 (0.07)	0.89 (0.19)	0.89 (0.11)
	MV	1.19 (0.04)	1.18 (0.05)	1.18 (0.05)	**1.23 (0.04)**	**1.21 (0.03)**	**1.20 (0.04)**
	CT	0.54 (0.02)	0.54 (0.01)	0.56 (0.01)	0.63 (0.01)	0.64 (0.01)	0.66 (0.04)
SMBPO	SR	0.94 (0.02)	0.88 (0.03)	0.81 (0.03)	0.85 (0.05)	0.67 (0.10)	0.60 (0.13)
	CR	0.05 (0.02)	0.07 (0.03)	0.14 (0.03)	0.12 (0.05)	0.29 (0.10)	0.35 (0.13)
	AV	1.00 (0.02)	1.01 (0.02)	1.00 (0.03)	0.89 (0.02)	0.88 (0.03)	0.88 (0.04)
	MV	1.18 (0.01)	1.18 (0.02)	1.19 (0.02)	1.21 (0.02)	1.20 (0.02)	1.20 (0.02)
	CT	0.55 (0.01)	0.56 (0.01)	0.56 (0.02)	0.66 (0.01)	0.68 (0.02)	0.68 (0.03)
MPC-CBF	SR	0.93 (0.01)	0.87 (0.03)	0.77 (0.05)	0.81 (0.03)	0.65 (0.08)	0.59 (0.12)
	CR	0.05 (0.01)	0.09 (0.03)	0.19 (0.05)	0.15 (0.03)	0.31 (0.08)	0.38 (0.12)
	AV	1.03 (0.03)	1.02 (0.03)	1.01 (0.05)	0.92 (0.03)	0.91 (0.05)	0.89 (0.09)
	MV	1.18 (0.01)	1.17 (0.02)	1.15 (0.03)	1.18 (0.03)	1.16 (0.03)	1.14 (0.06)
	CT	0.54 (0.01)	0.54 (0.02)	0.55 (0.02)	0.66 (0.02)	0.67 (0.03)	0.67 (0.04)
EIRM-RL	SR	**0.98 (0.00)**	**0.95 (0.01)**	**0.92 (0.02)**	**0.90 (0.04)**	**0.73 (0.08)**	**0.66 (0.11)**
	CR	**0.00 (0.00)**	**0.02 (0.01)**	**0.02 (0.02)**	**0.09 (0.04)**	**0.21 (0.08)**	**0.34 (0.11)**
	AV	1.00 (0.01)	0.99 (0.02)	0.97 (0.01)	0.89 (0.02)	0.86 (0.02)	0.80 (0.04)
	MV	1.19 (0.01)	1.19 (0.01)	1.18 (0.01)	1.20 (0.02)	1.19 (0.01)	1.18 (0.01)
	CT	0.55 (0.01)	0.55 (0.02)	0.57 (0.02)	0.68 (0.03)	0.69 (0.02)	0.69 (0.02)

2) Test evaluation: To comprehensively evaluate the robustness and generalization of the proposed EIRM-RL method, we conduct 100-episode test experiments in two scenarios under three noise levels. Here, Level-0 refers to adding zero-mean Gaussian noise to the position states, while Level-1 and Level-2 use Gaussian noise with different biases to simulate more challenging and realistic sensor disturbances. Notably, Level-2 represents an extreme case, mimicking sensor failure or severe observation corruption. The statistical test results are summarized in Table IV.

Specifically, in the static environment, EIRM-RL consistently achieves the highest task completion rates and the lowest collision rates under all noise levels. For example, under Level-2 noise, EIRM-RL attains a success rate of 0.92, which corresponds to improvements of 53.33%, 43.75%, 33.33%, 13.58%, and 19.48% compared to SAC, SAC-Lag, MBPO, SMBPO, and MPC-CBF, respectively. In terms of collision rate, EIRM-RL achieves 0.02, reducing the collision probability by up to 93.94% relative to SAC, by 85.71% compared to SMBPO, and by 89.47% relative to MPC-CBF. For efficiency metrics, EIRM-RL maintains average and maximum velocities (0.97 and 1.18, respectively) comparable to all baselines, and the computing time remains stable at 0.57 seconds per episode, on par with or better than the other methods.

In the dynamic environment, which poses greater challenges due to moving obstacles and more complex interactions, EIRM-RL continues to demonstrate robust performance. Under Level-2 noise, EIRM-RL achieves a success rate of 0.66, outperforming SAC, SAC-Lag, MBPO, SMBPO, and MPC-CBF by 15.79%, 10.00%, 17.86%, 10.00%, and 11.86%, respectively. The collision rate of EIRM-RL in this adversarial setting is 0.34, which is 48.48% lower than SAC, 2.86% lower than SMBPO, and 10.53% lower than MPC-CBF. Across all noise levels, EIRM-RL's average and maximum velocities remain stable, and the computing time is consistently low at 0.69 seconds per episode.

In summary, EIRM-RL consistently outperforms all baselines in both static and dynamic environments across varying noise levels. Notably, the Level-2 disturbance introduces sensor deviations beyond the training range, providing a direct evaluation of the model's extrapolation robustness. Under this most challenging condition, EIRM-RL achieves the highest success rate (0.92 static, 0.66 dynamic), improving over SAC by 53.33% and 15.79%, and over SMBPO by 13.58% and 10.00%, and over MPC-CBF by 19.48% and 11.86%, respectively. The collision rate is significantly reduced to 0.02 (static) and 0.34 (dynamic), yielding relative reductions of up to 93.94% and 48.48% compared to SAC. These results demonstrate that EIRM-RL maintains high success and low collision rates even under out-of-distribution noise, indicating

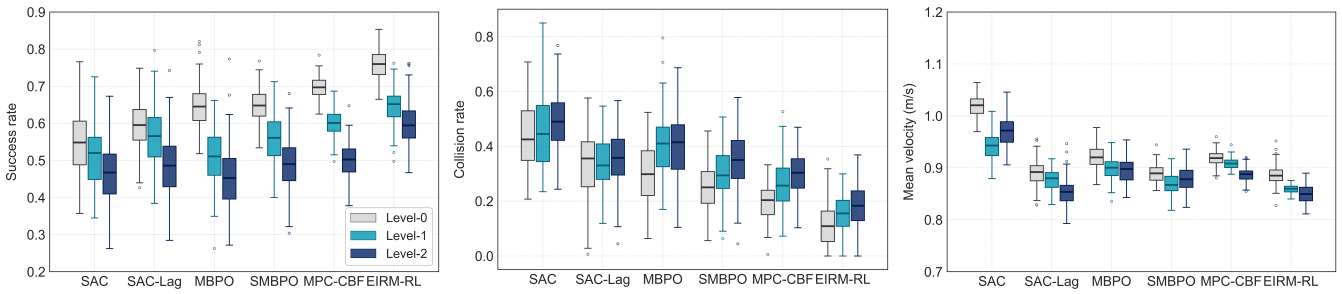


Fig. 7. Comprehensive test comparison under different noise levels in scenario (c), showing success rate (left), collision rate (middle), and mean velocity (right).

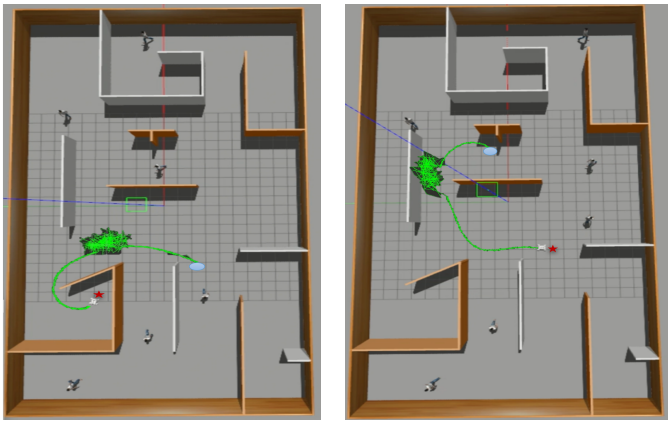


Fig. 8. Representative trajectories of UGV navigation in the unseen maze scenario (c). The blue ellipse marks the start point, the red star denotes the goal point, and the green curve represents the position states received by the UGV during navigation. (a) shows a successful path through narrow corridors, while (b) demonstrates adaptive obstacle avoidance under noisy observations.

strong generalization and practical reliability in handling unexpected sensor disturbances. Furthermore, the modular design of EIRM-RL enables its straightforward extension to diverse robot types, sensor modalities, and application scenarios, demonstrating robust versatility and adaptability for a wide range of safety-critical robotic tasks.

Moreover, EIRM-RL maintains competitive motion efficiency with average/maximum speeds of 0.97/1.18 m/s, respectively and a small standard deviation, whilst preserving real-time performance with per-round computation time stabilising between 0.57-0.69 seconds. Given that each test round comprises 200 decision steps, this translates to an average computational time per step of approximately 2.85 milliseconds (static, level-2) to 3.45 milliseconds (dynamic, level-2). This demonstrates that the proposed risk assessment and alert mechanism is efficiently integrated into the policy inference process, with the entire system meeting the real-time operational requirements on practical robotic platforms.

3) Generalization test evaluation: To further validate the generalization capability of the proposed EIRM-RL algorithm beyond the training distribution, we conduct zero-shot evaluations in the unseen scenario (c), as illustrated in Fig. 3. In this evaluation, the policy trained exclusively in the static and dynamic scenarios is directly deployed to scenario (c) without any retraining or fine-tuning. For each comparison method, we execute 20 independent tests across three observation noise levels. The comprehensive comparison results are shown in Fig. 7.

Across all noise levels, the proposed EIRM-RL maintains the highest overall success rate and lowest collision probability. Under the most challenging Level-2 noise conditions, EIRM-RL achieves a success rate of 0.61-representing improvements of 20.45%, 14.04%, 18.18%, 10.11%, and 9.29% over SAC, SAC-Lag, MBPO, SMBPO, and MPC-CBF, respectively. The collision rate drops to 0.33, representing reductions of 46.03%, 28.26%, and 10.81% compared to SAC, SAC-Lag, MBPO, SMBPO, and MPC-CBF, respectively. Concurrently, EIRM-RL maintains motion efficiency with average and peak velocities of 0.92 m/s and 1.12 m/s, demonstrating stability and safety in unstructured environments. Furthermore, the narrow distribution in Fig. 7 indicates significantly reduced performance fluctuations across varying noise levels, suggesting enhanced stability of the learned policy. This consistency reveals that EIRM-based epistemic integrity monitoring and safe constraint effectively regularize the learned representations, enabling the policy to adapt to unseen spatial patterns and random perturbations.

Fig. 8 illustrates typical navigation trajectories of UGV under state perturbations in an unseen maze environment. As shown in Fig. 8(a), EIRM-RL successfully traverses narrow passages and executes smooth turns to reach the goal. In Fig. 8(b), the algorithm adapts to the environment despite moving obstacles and local occlusions, maintaining a stable trajectory even under noisy interference. These visualizations clearly demonstrate the proposed method's effective perception correction and risk-aware decision-making capabilities.

In summary, additional testing in an unseen scenario confirms that the proposed EIRM-RL method not only excels in the training domain but also exhibits strong generalization capabilities. These results validate the method's robustness, proving its ability to transfer the learned safety-performance tradeoff to entirely new, untrained environments.

4) Ablation study: To systematically evaluate the individual contributions of each component within the proposed EIRM-RL framework, we conduct goal-oriented ablation studies grounded in the algorithmic structure outlined in Section II.

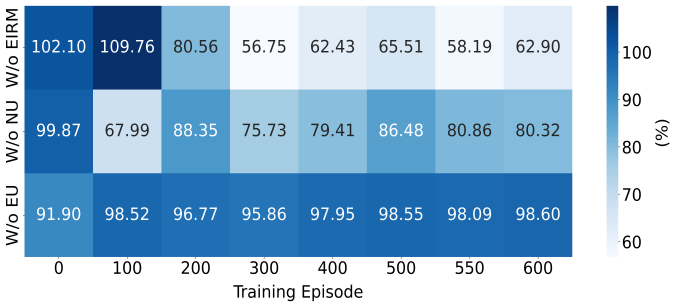


Fig. 9. Ablation study results, where the metric number represents the rate of the average value of ablation candidate relative to the EIRM-RL at the same training stage.

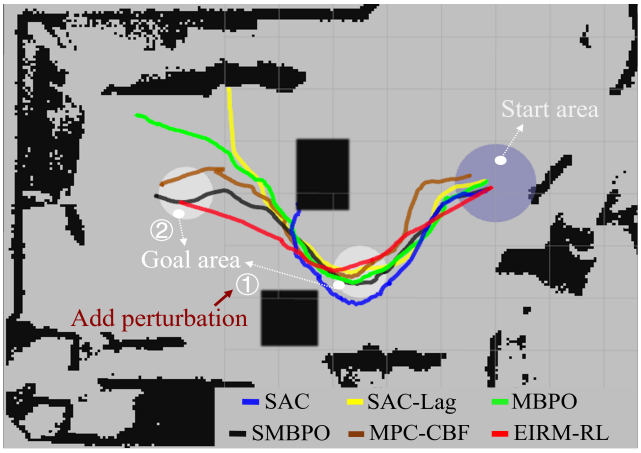


Fig. 10. Trajectory comparison of different algorithms in the real-world static environment.

Specifically, the following variants are considered: W/o EIRM, which eliminates the EIRM module and trains the agent purely for task rewards; W/o NU, which excludes noise related uncertainty, utilizing only the world model latent cost c_t and the noise uncertainty σ^2_{noise}; and W/o EU, which removes the epistemic variance term σ^2_{epist} from both the extended PL formulation and subsequent policy optimization.

The ablation results are shown in Fig. 9, where each unit shows the average reward ratio of each variant relative to EIRM-RL at the same training stage. The results show that W/o EIRM leads to the most severe performance degradation, with performance drops of up to 37.10%, and an average loss of more than 30.10% throughout training. Excluding W/o NU causes a moderate reduction, with losses between 13.52% and 19.43% and an average of 16.30%. In contrast, W/o EU leads to the smallest performance drops, averaging 3.35%, but still has a noticeable impact on the robustness of the final policy. These findings confirm that EIRM is critical to ensuring safety and efficiency, and comprehensive multi-source uncertainty modeling is essential to achieving robust and generalizable performance. In summary, the ablation study confirms that each component has an indispensable contribution in the safety, robustness, and generalization of EIRM-RL.

B. Real-world Experiments

1) Static environment: To evaluate the effectiveness and robustness of the proposed EIRM-RL algorithm in real-world scenarios, we conducted trajectory comparison experiments in a static environment with multiple obstacles. As shown in Fig. 10, the UGV needs to autonomously navigate from the starting area to two consecutive goal points while avoiding static obstacles. To further test the adaptability of each method, a position perturbation is introduced after reaching the first goal point, which is modeled as a Gaussian noise of $\mathcal{N}(2, 0.5^2)$. This strong perturbation simulates the worst-case sensor failure observation, forcing the UGV to replan its trajectory towards the second goal point.

The experimental results show the representative trajectories generated by different algorithms, including SAC, SAC-Lag, MBPO, SMBPO, and the proposed EIRM-RL. Compared with other baseline methods, EIRM-RL shows a more stable and efficient navigation path, successfully reaches the two goal areas, and shows excellent robustness to the introduced perturbations. These results prove that even under worst-case conditions, EIRM-RL achieves higher adaptability and reliability in multi-stage real-world navigation tasks. Furthermore, compared with the model-based method MPC-CBF, the trajectory generated by MPC-CBF shows cautious detour behavior around obstacles, resulting in longer paths and delayed arrival times, which is more conservative than the EIRM-RL. While MPC-CBF ensures strict safety margins, its predefined constraint set renders it insufficiently adaptive to sudden perceptual disturbances, leading to trajectory deviations when positional noise is applied. In contrast, EIRM-RL responds to environmental changes by dynamically adjusting the control strategy, effectively balancing safety and efficiency. Its generated trajectories exhibit smoother curvature transitions and faster target convergence while maintaining safety compliance in densely obstructed areas. These results demonstrate that EIRM-RL significantly enhances adaptability and reliability to real-world uncertainties while preserving the safety properties of model-based control.

2) Dynamic environment: In addition to the quantitative evaluation conducted in the static environment, we further verify the effectiveness of our proposed method in a dynamic environment, where both static and dynamic obstacles are present. In this scenario, dynamic obstacles are introduced by human participants walking randomly within the workspace, as shown in Fig. 4(b). To provide an intuitive and comprehensive demonstration, the test results in the dynamic environment are presented qualitatively in the supplementary video (see part 5 of the video: https://youtu.be/khBhRrMxDcc).

As captured in the video, the UGV is able to successfully avoid both static obstacles and oncoming pedestrians, and ultimately reaches the specified goal area. These qualitative results further validate the adaptability and robustness of the EIRM-RL method in complex, real-world dynamic environments.

V. Conclusion

This paper proposes EIRM-RL, a novel SMBRL framework that integrates epistemic integrity risk monitoring for safe and

robust mapless navigation under uncertainty. The EIRM-RL combines an ensemble world model with explicit estimation of multi-source uncertainties. EIRM-RL dynamically calculates an extended PL for real-time risk assessment and enforces safety constraints during policy optimization. This principled method enables the agent to proactively avoid hazards and maintain trustworthy navigation performance even under severe observation disturbances. Extensive simulations in both static and dynamic environments demonstrate that EIRM-RL consistently outperforms state-of-the-art baselines in terms of mission success rate, collision rate, and cumulative reward, especially under challenging sensor noise and fault conditions. Ablation studies highlight the critical role of risk monitoring and comprehensive uncertainty modeling in enhancing safety, robustness, and generalization. Furthermore, real-world experiments validate the practical effectiveness of the framework, where EIRM-RL enables a UGV to reliably reach its target and avoid obstacles even in the presence of dynamic hazards and sensor perturbations.

Although our method demonstrates promising results in trustworthy autonomous navigation, some limitations remain. The current approach assumes Gaussian sensor noise, which may not fully capture real-world disturbances. In addition, the framework requires further validation in highly unstructured, crowded, or dynamic multi-agent environments. Future work will address these challenges by incorporating non-Gaussian noise and adversarial perturbation models, and applying advanced representation learning methods such as attention-based multimodal sensor fusion. These efforts aim to further improve the robustness and practical applicability of EIRM-RL in real-world scenarios.

REFERENCES

[1] D. Hu, G. Zhou, J. Wu, and C. Huang, "Trust-calibrated human-in-the-loop reinforcement learning for safe and efficient autonomous navigation," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[2] C. Li, H. Zhang, K. Chen, and M. Yang, "Novel noise-tolerant zeroing neurodynamics algorithms for dynamic nonlinear least square problems with robot application," *IEEE Transactions on Industrial Electronics*, pp. 1–10, 2025.

[3] Q. Zhang, X. Niu, and C. Shi, "Impact assessment of various imu error sources on the relative accuracy of the gnss/ins systems," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 5026–5038, 2020.

[4] J. Tan, J. Wan, B. Chen, M. Safran, S. A. AlQahtani, and R. Zhang, "Selective feature reinforcement network for robust remote fault diagnosis of wind turbine bearing under non-ideal sensor data," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[5] V. M. Zavala and L. T. Biegler, "The advanced-step nmpc controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, 2009.

[6] I. S. Mohamed, J. Xu, G. S. Sukhatme, and L. Liu, "Toward efficient mppi trajectory generation with unscented guidance: U-mppi control strategy," *IEEE Transactions on Robotics*, vol. 41, pp. 1172–1192, 2025.

[7] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1313–1325, 2021.

[8] K. Shen, Y. Li, T. Liu, J. Zuo, and Z. Yang, "Adaptive-robust fusion strategy for autonomous navigation in gnss-challenged environments," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6817–6832, 2024.

[9] Y. Zhang, Y. Wang, P. Yan, and W. Wen, "Learning safe, optimal, and real-time flight interaction with deep confidence-enhanced reachability guarantee," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2025.

[10] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3679–3685.

[11] J. Wu, H. Yang, L. Yang, Y. Huang, X. He, and C. Lv, "Human-guided deep reinforcement learning for optimal decision making of autonomous vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 11, pp. 6595–6609, 2024.

[12] J. Wu, C. Huang, H. Huang, C. Lv, Y. Wang, and F.-Y. Wang, "Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 164, p. 104654, 2024.

[13] Q. Zhou, Y. Niu, W. Xiang, and L. Zhao, "A novel reinforcement learning algorithm based on broad learning system for fast communication antijamming," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 3, pp. 2590–2599, 2025.

[14] D. Lee and M. Kwon, "Episodic future thinking with offline reinforcement learning for autonomous driving," *IEEE Internet of Things Journal*, vol. 12, no. 11, pp. 17 012–17 023, 2025.

[15] D. Hu, H. Xie, K. Song, Y. Zhang, and L. Yan, "An apprenticeship-reinforcement learning scheme based on expert demonstrations for energy management strategy of hybrid electric vehicles," *Applied Energy*, vol. 342, p. 121227, 2023.

[16] Z. Gao, H. Hao, F. Gao, and R. Zhao, "Constrained reinforcement-learning-enabled policies with augmented lagrangian for cooperative intersection management," *IEEE Internet of Things Journal*, vol. 12, no. 5, pp. 5396–5411, 2025.

[17] J. Wu, Y. Zhou, H. Yang, Z. Huang, and C. Lv, "Human-guided reinforcement learning with sim-to-real transfer for autonomous navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 745–14 759, 2023.

[18] D. M. Bossens, "Robust lagrangian and adversarial policy gradient for robust constrained markov decision processes," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 1227–1239.

[19] Q. Liu, Y. Li, X. Shi, K. Lin, Y. Liu, and Y. Lou, "Distributional policy gradient with distributional value function," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[20] Y. Zhang, W. Wen, and P. Yan, "Safe-assured learning-based deep se(3) motion joint planning and control for uav interactions with dynamic environments," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, 2024, pp. 4222–4229.

[21] D. Hu, C. Huang, J. Wu, and X. Yuan, "Toward multi-task generalization in autonomous navigation: A human-in-the-loop adversarial reinforcement learning with diffusion policy," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2025.

[22] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *Advances in neural information processing systems*, vol. 30, 2017.

[23] S. Safaoui, A. P. Vinod, A. Chakrabarty, R. Quirynen, N. Yoshikawa, and S. D. Cairano, "Safe multiagent motion planning under uncertainty for drones using filtered reinforcement learning," *IEEE Transactions on Robotics*, vol. 40, pp. 2529–2542, 2024.

[24] A. Romero, Y. Song, and D. Scaramuzza, "Actor-critic model predictive control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 777–14 784.

[25] W. Huang, Y. Cui, H. Li, and X. Wu, "Practical probabilistic model-based reinforcement learning by integrating dropout uncertainty and trajectory sampling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 7, pp. 12 812–12 826, 2025.

[26] Y. Cui, W. Shi, H. Yang, C. Shao, L. Peng, and H. Li, "Probabilistic model-based reinforcement learning unmanned surface vehicles using local update sparse spectrum approximation," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1283–1293, 2024.

[27] X. He, J. Wu, Z. Huang, Z. Hu, J. Wang, A. Sangiovanni-Vincentelli, and C. Lv, "Fear-neuro-inspired reinforcement learning for safe autonomous driving," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 267–279, 2024.

[28] H.-L. Hsu, H. Meng, S. Luo, J. Dong, V. Tarokh, and M. Pajic, "Reforma: Robust reinforcement learning via adaptive adversary for drones flying under disturbances," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5169–5175.

[29] X. Zhang, X. Cai, B. Liu, W. Huang, S.-C. Zhu, S. Qi, and Y. Yang, "Differentiable information enhanced model-based reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 21, 2025, pp. 22 605–22 613.

[30] M. A. Sturza, "Navigation system integrity monitoring using redundant measurements," *Navigation*, vol. 35, no. 4, pp. 483–501, 1988.

[31] X. Xia, W. Wen, and L.-T. Hsu, "Integrity-constrained factor graph optimization for gnss positioning in urban canyons," *NAVIGATION: Journal of the Institute of Navigation*, vol. 71, no. 3, 2024.

[32] H. Jing, Y. Gao, S. Shahbeigi, and M. Dianati, "Integrity monitoring of gnss/ins based positioning systems for autonomous vehicles: State-of-the-art and open challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 166–14 187, 2022.

Yuanyuan Zhang received the B.E. degree in Traffic and Transportation Engineering from the Shandong University, Jinan, China, in 2020, and the M.S. degree in Power Engineering from the Tianjin University, Tianjin, China, in 2023. She is currently working toward the Ph.D. degree with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong. Her research interests include trustworthy navigation, adaptive control, deep reinforcement learning, and robotics.

Yingying Wang received the B.E. degree in Electronic Engineering from Northeastern University, Shenyang, Liaoning, China, in 2016, and the M.S. degree in Signal Processing from Northeastern University, Shenyang, Liaoning, China, in 2019. She received the PhD degree in the Department of Electronic Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong SAR, China, in 2023. She was a postdoctoral researcher in the Hong Kong Automotive Platforms and Application Systems (APAS) R&D center from 2023.8 to 2024.7. She is a postdoctoral fellow in the Intelligent Positioning and Navigation Lab at The Hong Kong Polytechnic University, Hong Kong SAR, China. Her research interests are pedestrian localization and non-intrusive intelligent sensing.

[33] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[34] D. Hu and Y. Zhang, "Deep reinforcement learning based on driver experience embedding for energy management strategies in hybrid electric vehicles," *Energy Technology*, vol. 10, no. 6, p. 2200123, 2022.

[35] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," *Advances in neural information processing systems*, vol. 32, 2019.

[36] W. Huang, Y. Zhou, X. He, and C. Lv, "Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1832–1845, 2024.

[37] Q. Yang, T. D. Simão, S. H. Tindemans, and M. T. Spaan, "Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 639–10 646.

[38] Z. Xu, B. Liu, X. Xiao, A. Nair, and P. Stone, "Benchmarking reinforcement learning techniques for autonomous navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9224–9230.

Weisong Wen (Member, IEEE) received a BEng degree in Mechanical Engineering from Beijing Information Science and Technology University (BISTU), Beijing, China, in 2015, and an MEng degree in Mechanical Engineering from the China Agricultural University, in 2017. After that, he received a Ph.D. degree in mechanical engineering, the Hong Kong Polytechnic University. He was a visiting student researcher at the University of California, Berkeley (UCB) in 2018. He is currently an assistant professor in the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University. His research interests include multi-sensor integrated localization for autonomous vehicles, SLAM, and GNSS positioning in urban canyons.