Trustworthy Cyber-Resilient Reinforcement Learning for Secure Navigation under Adversarial Attack

Ashwin Devanga, *Student Member, IEEE,* Tingjun Lei*, *Member, IEEE,* Jueming Hu, *Member, IEEE*, Jun Chen, *Senior Member, IEEE*, and Chaomin Luo, *Senior Member, IEEE*

***Abstract*—The rapid deployment of autonomous ground vehicles (AGVs) across critical domains has heightened concerns regarding their susceptibility to cyber–physical threats. In particular, adversarial obstacle injection, whereby false objects are maliciously introduced into sensor data, poses a severe risk to navigation safety and reliability. To address this challenge, a cyber-resilient reinforcement learning framework is developed based on the proposed Enhanced Deep Deterministic Policy Gradient (EDDPG) algorithm. The framework unifies path planning and control through continuous action outputs and incorporates an adversarial training strategy in which the agent is deliberately exposed to deceptive sensor inputs during training. By learning to differentiate between legitimate hazards and falsified obstacles, the trained policy acquires resilience to sensor manipulation while maintaining efficient and safe navigation. Comparative studies against conventional planners and baseline reinforcement learning models indicate that the adversarially trained EDDPG agent consistently achieves more robust trajectory performance under attack conditions. This work demonstrates that resilience in autonomous navigation must be explicitly engineered and provides a practical methodology for constructing secure and trustworthy AGV navigation systems.**

***Index Terms*—Reinforcement Learning, Autonomous Systems, Path Planning, Obstacle Avoidance, Adversarial Obstacle Injection, Man-in-the-Middle.**

2025 Cyber Awareness and Research Symposium (CARS) | 979-8-3315-9628-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CARS67163.2025.11337781

I. Introduction

THE deployment of Autonomous Ground Vehicles (AGVs) is rapidly expanding across critical sectors, including logistics, public transportation, and agriculture. The ability of these vehicles to navigate complex environments without human intervention is crucial for their successful operation [1]–[4]. However, the increasing reliance on sensor data from sources like LiDAR and cameras introduces significant cybersecurity vulnerabilities [5]–[7]. This paper addresses the pressing challenge of ensuring navigational resilience against adversarial sensor attacks, where malicious data manipulation can compromise vehicle safety and mission success.

Traditional path planning algorithms have been extensively utilized in autonomous navigation tasks, including graph-based frameworks, swarm intelligence techniques, tree-based models, and neural networks [8]–[13]. In graph-based methods, researchers have proposed bio-inspired approaches that integrate optimal routing in cluttered environments [14], while others have developed node-selection algorithms using Voronoi diagrams to ensure safety-aware navigation [15]. Bio-inspired algorithms have also been popular, with hybrid bat-pigeon-inspired models designed for vehicle navigation [16] and bat algorithms applied to image-based path planning [17]. Furthermore, advanced frameworks have utilized brainstorm optimization for multi-objective navigation [18] and informed sampling strategies to improve exploration efficiency [19].

Although these methods can effectively generate collision-free trajectories in predictable settings, they fundamentally assume the integrity of the sensory input they receive [20]. This makes them highly susceptible to deception; an attacker can introduce false obstacles into the vehicle's perception data, causing these algorithms to generate inefficient or dangerously diverted trajectories. Additionally, many of these approaches decouple the planning and control processes, posing challenges for real-time, integrated navigation in dynamic domains. This kind of an attack is known as Adversarial Obstacle Injection (AOI) [21]. While there have been efforts to mitigate this type of attack [22], [23], our proposed method can inherently resist such attacks through proper training.

In recent years, deep reinforcement learning (DRL) has emerged as a promising alternative for creating integrated navigation and control systems [24]. DRL enables an agent to learn optimal control policies by interacting directly with its environment, unifying perception, decision-making, and control within a single framework. In the context of autonomous vehicles, DRL has been applied successfully to path planning under environmental disturbances [25] and for robust obstacle avoidance under partially observable conditions [26]. While these DRL approaches demonstrate strong adaptability, a conventionally trained agent remains vulnerable to adversarial attacks. If not explicitly trained to handle data manipulation, it will treat a malicious non-existent obstacle as a legitimate one, compromising its mission just as a traditional algorithm would. The core problem, therefore, is not just creating an intelligent navigation agent, but creating one that is inherently resilient to deception.

To overcome this limitation, this study introduces an Enhanced Deep Deterministic Policy Gradient (EDDPG) framework that achieves cyber-resilience through adversarial training. The EDDPG algorithm is well-suited for AGV control as it directly outputs continuous steering and throttle commands, enabling smooth and precise vehicle motion. The central innovation is a training regimen where the DRL agent is

A. Devanga and T. Lei are with the School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA. J. Hu is with the Department of Mechanical Engineering, University of North Dakota, Grand Forks, ND 58202, USA. J. Chen is with the Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309 USA. C. Luo is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762, USA. (Corresponding Author: T. Lei, Tingjun.Lei@UND.edu)

intentionally exposed to simulated Adversive Obstacle Injection attacks. By learning in an environment where its senses cannot always be trusted, the agent develops a robust policy that can differentiate between legitimate hazards and malicious artifacts. The key contributions of this work are as follows:

(1) An EDDPG model tailored for real-time AGV navigation that unifies path planning and control by directly outputting continuous throttle and steering commands.
(2) The integration of an adversarial training paradigm to build resilience against AOI attacks, enhancing the security of the navigation system.
(3) A comparative analysis demonstrating that the adversarially trained agent maintains superior mission performance and trajectory efficiency when under attack compared to conventionally trained agents.

II. Problem Formulation: AGV Navigation Framework

This study addresses the critical cyber-physical security vulnerabilities inherent in autonomous vehicle navigation as shown in Fig. 1. The core problem is framed as securing an AGV against malicious data manipulation designed to compromise its mission. To this end, we define the navigation task within a reinforcement learning context, detailing the specific threat model, the system under attack, and the physical dynamics that an adversary seeks to exploit.

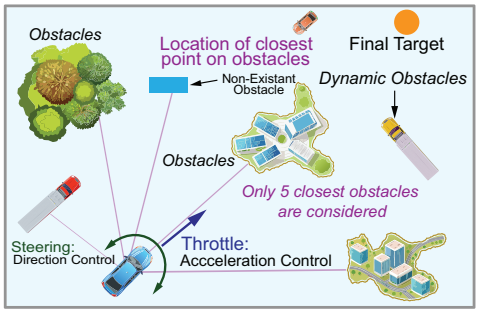

Fig. 1. Illustration of the AGV navigation problem in a dynamic environment. The AGV is controlled via continuous throttle (acceleration) and steering (direction) commands while subject to spoofed obstacles. Both static and dynamic obstacles are present, and only the five closest obstacles are explicitly considered in the state representation. AOI adds unidentified and non-existent obstacles into the environment.

A. Cybersecurity Threat Modeling

The increasing reliance of AGVs on sensor data from sources like LiDAR and cameras creates a significant attack surface. Our threat model is built on the premise that a malicious actor can compromise the data link between the AGV's perception sensors and its navigation module. This vulnerability is exploited through a Man-in-the-Middle (MitM) attack, which allows the adversary to intercept and alter the data stream without the agent's knowledge.

The specific attack vector we focus on is Adversarial Obstacle Injection (AOI), a form of data integrity corruption. In an AOI attack, the adversary's objective is not to cause a catastrophic system failure, which would be easily detected, but rather to subtly degrade mission performance. This is achieved by injecting the coordinates of non-existent virtual obstacles into the vehicle's state representation, S_t. These malicious obstacles are strategically placed along the AGV's optimal path, forcing the navigation agent to compute inefficient and diverted trajectories as it performs evasive maneuvers. By continuously manipulating the perceived environment, the attacker can effectively control the AGV's path, increasing travel time and energy consumption, and ultimately undermining mission success. The operational area of the environment contains static rectangular obstacles with side lengths L_{ki} sampled uniformly within

$$L_{ki} = \text{rand}(0,1) \cdot (S_{\max\ \text{obs}} - S_{\min\ \text{obs}}) + S_{\min\ \text{obs}}, \quad (1)$$

where $S_{\min\ \text{obs}} = 0.5$ m and $S_{\max\ \text{obs}} = 3.0$ m. Obstacle positions are randomized at the beginning of each episode, ensuring variability in difficulty and preventing overfitting to specific layouts. The navigation objective is defined by a target coordinate $(x_{\text{target}}, y_{\text{target}})$, with successful arrival declared if the AGV enters a tolerance radius r_{target}. An episode terminates if the AGV reaches the target, collides with an obstacle, exits the operational domain, or becomes trapped in a cyclic trajectory. These termination conditions collectively enforce mission feasibility and ensure that the learned policy remains robust across diverse and dynamic scenarios. The episode does not end if the agent collides with a spoofed obstacle. It continues on as though the obstacle does not exist.

B. Markov Decision Process Formulation

The AGV navigation problem is modeled as an MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(S_{t+1} \mid S_t, A_t)$ denotes the transition dynamics, $\mathcal{R}(S_t, A_t, S_{t+1})$ defines the task rewards, and $\gamma \in (0,1]$ discounts future returns. The objective is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected discounted return,

$$\pi^\star = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right], \quad (2)$$

thereby unifying perception, decision-making, and low-level actuation within a single optimization framework appropriate for real-time control in uncertain, dynamic conditions.

At each decision epoch, the environment provides a state vector S_t that encapsulates environmental context and vehicle kinematics. The environmental component encodes the global coordinates $(x_{\text{edge}}, y_{\text{edge}})$ of the five closest obstacle edge points relative to the AGV, together with the target location $(x_{\text{target}}, y_{\text{target}})$ as shown in Fig 1. The kinematic component comprises the AGV's position (x_t, y_t), heading h_t and linear speed v_t. All quantities are concatenated and flattened into a single vector, yielding a compact yet informative representation that supports stable policy learning and efficient inference. The action space is continuous and two-dimensional, $A_t = [\, a_{\text{throttle}},\, a_{\text{steering}} \,]^\top$ with $a_{\text{throttle}}, a_{\text{steering}} \in [-1, 1]$. The throttle command regulates forward or reverse acceleration, where $a_{\text{throttle}} = 1$ indicates maximum forward acceleration and $a_{\text{throttle}} = -1$ indicates maximum reverse. The steering command modulates turning, with $a_{\text{steering}} = \pm 1$ corresponding to maximum rates in opposite directions. This continuous

parameterization enables learning smooth, fine-grained control policies that are better aligned with continuous actuation than discretized action sets.

C. AGV Kinematic and Dynamic Model

The AGV motion model captures both translational and rotational dynamics under bounded actuation. The AGV's state is represented by its global position (x_t, y_t), forward velocity v_t, heading h_t, and angular velocity ω_t. The throttle input $a_{\text{throttle}} \in [-1, 1]$ determines the linear acceleration, scaled by a maximum $A_{\max} = 0.8\ m/s^2$. The resulting effective acceleration is given by $a_t = a_{\text{throttle}} \cdot A_{\max}$. The velocity update follows $v'_{t+1} = v_t + a_t \Delta t$, and is then clipped to the admissible operational range: $v_{t+1} = \text{clip}(v'_{t+1}, v_{\min}, v_{\max})$, with $v_{\min} = 0.3$ m/s and $v_{\max} = 1.0$ m/s. The steering input $a_{\text{steering}} \in [-1, 1]$ modulates angular velocity.

$$\omega'_{t+1} = \frac{v_t}{L} \tan(a_{\text{steering}}.\delta_{max}), \tag{3}$$

where L is the length of the wheelbase of the AGV and δ_{max} is the maximum steering angle allowed on the vehicle. δ_{max} reduces with increased speed. Heading is updated using the trapezoidal integration rule to reduce numerical error,

$$h_{t+1} = \left(h_t + \frac{\omega_t + \omega_{t+1}}{2} \Delta t \right) \bmod 360, \tag{4}$$

and the position is propagated based on the average velocity and heading across the time interval. This formulation provides a physics-consistent representation of AGV dynamics, ensuring that learning-based control policies respect realistic maneuverability and actuation constraints. The control systems also do not have the ability to differentiate between spoofed and real obstacles.

III. Cyber-Resilient EDDPG Framework

To counter the threat of AOI, we have developed a specialized defensive architecture based on an EDDPG framework. This framework is engineered not just for efficient navigation but for robust operation within a compromised sensory environment. Its core purpose is to train a control policy that can withstand malicious data manipulation. Built upon the actor-critic architecture of DDPG, the framework directly outputs continuous throttle and steering commands, enabling smooth, real time trajectory execution without the need for a separate low-level controller. The enhancement over standard DDPG lies in the integration of OU noise for exploration-stabilized learning, coupled with a high-fidelity environment model that captures real and spoofed obstacle interactions. The architecture, shown in Fig. 2, integrates a secure control mechanism with a specialized adversarial training regimen to produce an agent that is inherently resilient to deception.

The EDDPG framework employs two neural networks: an actor $\mu(S|\theta^\mu)$ that deterministically maps a state vector to a continuous action, and a critic $Q(S, A|\theta^Q)$ that estimates the state-action value function. Both networks are accompanied by target networks μ' and Q', which are updated using a soft-update mechanism with rate $\tau \ll 1$ to improve training stability. The actor learns a deterministic policy that maximizes the critic's estimated return, while the critic is trained to minimize the temporal-difference error using the Bellman equation. The deterministic formulation eliminates the stochastic sampling variance associated with policy-gradient methods in continuous spaces, allowing for more precise control outputs that are critical for marine vehicle actuation. As illustrated in Fig. 2, the EDDPG framework employs both online and target networks for the actor and critic, ensuring stable updates through soft target updates.

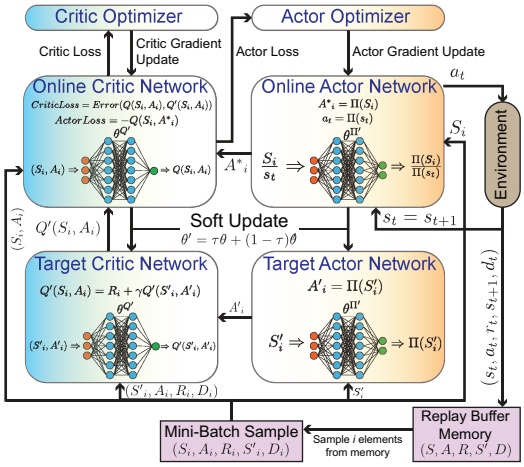

Fig. 2. Architecture of the proposed EDDPG framework for AGV navigation. The framework follows an actor-critic structure with online and target networks for both actor and critic. The online actor generates continuous control actions (steering and throttle) based on the observed state, while the online critic evaluates their expected returns. Target networks are updated via soft updates to stabilize training. A replay buffer stores past transitions, from which mini-batches are sampled for training. Gradient updates are applied separately to the actor and critic through their respective optimizers.

Effective exploration in continuous action spaces requires temporally correlated perturbations that reflect the inertia and persistence of physical control signals. To achieve this, the EDDPG framework adds OU noise N_t^{OU} to the actor's actions during training. The OU process is defined as:

$$N_{t+\Delta t}^{OU} = N_t^{OU} + \theta(\mu_{\text{noise}} - N_t^{OU})\Delta t + \sigma_{\text{noise}} \sqrt{\Delta t}\, \mathcal{W}_t, \tag{5}$$

where θ controls the mean reversion speed, μ_{noise} is the mean value, σ_{noise} is the volatility, and $\mathcal{W}_t \sim \mathcal{N}(0, 1)$ is Gaussian noise. This formulation produces smooth, correlated exploration that accelerates convergence and yields more stable post-convergence performance, particularly in environments with continuous actuation like throttle and steering control. This also means the perturbations are smooth and persistent over time, effectively mimicking an adversary who is continuously updating a phantom obstacle's position relative to the AGV. By forcing the agent to train against these correlated adversarial signals, we build a policy that learns to distinguish between the persistent patterns of a malicious injection and the characteristics of a true physical obstacle, making it far more robust than an agent trained with simple random exploration.

The framework maintains an experience replay buffer $\mathcal{M}$ containing tuples $(S_t, A_t, R_t, S_{t+1}, d_t)$, where d_t is a terminal flag. To improve sample efficiency, the EDDPG implementation optionally employs Prioritized Experience Replay (PER),

in which transitions with higher temporal-difference errors are sampled with greater probability. This prioritization focuses updates on informative experiences, speeding up convergence while retaining sufficient diversity to prevent overfitting to a narrow subset of trajectories.

Algorithm 1 Proposed EDDPG Algorithm

Initialize critic $Q(S, A|\theta^Q)$, actor $\pi(S|\theta^\pi)$; targets Q', π' with $\theta^{Q'} \leftarrow \theta^Q, \theta^{\pi'} \leftarrow \theta^\pi$.
Initialize replay buffer M (capacity N_{buf}); noise $\mathcal{N}_{OU}$; hyperparameters: batch size K, γ, τ
for episode = 1 to $M_{episodes}$ **do** // Loop through training episodes
  Reset $\mathcal{N}_{OU}$ process
  Get initial state S_1
  for $t = 1$ to T_{max_steps} **do** // Loop through current episode
    Select action with Noise$A_t = \pi(S_t|\theta^\pi) + \mathcal{N}_{OU}(t)$
    Execute A_t; observe reward R_t, next state S_{t+1}, done flag d_t
    Store transition $(S_t, A_t, R_t, S_{t+1}, d_t)$ in B
    if $|B| \geq K$ **then**
      Sample random mini-batch of size K from B: $(S_i, A_i, R_i, S_{i+1}, d_i)$
      $A'_{i+1} = \pi'(S_{i+1}|\theta^{\pi'})$ // Expected action using target Actor
      $Q_i = R_i + \gamma(1 - d_i)Q'(S_{i+1}, A'_{i+1}|\theta^{Q'})$ // Expected Q value using target Critic
      $L(\theta^Q) = \frac{1}{K}\sum_i (Q_i - Q(S_i, A_i|\theta^Q))^2$. // Critic Loss
      ${A^*}_i = \pi(S_i|\theta^\pi)$
      $\nabla_{\theta^\pi} J \approx \frac{1}{K}\sum_i \left[\nabla_{A^*} Q(S_i, {A^*}_i|\theta^Q) \cdot \nabla_{\theta^\pi} \pi(S_i|\theta^\mu)\right]$ // Calculate Actor Gradients
      $\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$ // Update Target Critic
      $\theta^{\pi'} \leftarrow \tau\theta^\pi + (1 - \tau)\theta^{\mu'}$ // Update Target Actor
    end if
    $S_t \leftarrow S_{t+1}$ // Update current state
    if d_t **then**
      Break // Stop episode if done
    end if
  end for
end for

At each update step, the critic parameters θ^Q are optimized to minimize the mean-squared temporal-difference error:

$$\mathcal{L}(\theta^Q) = \frac{1}{K}\sum_{i=1}^{K}\left(y_i - Q(S_i, A_i|\theta^Q)\right)^2, \tag{6}$$

where

$$y_i = R_i + \gamma(1 - d_i)Q'(S_{i+1}, \mu'(S_{i+1}|\theta^{\mu'})|\theta^{Q'}). \tag{7}$$

The actor parameters θ^μ are updated by ascending the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{K}\sum_{i=1}^{K} \nabla_A Q(S_i, A|\theta^Q)\big|_{A=\mu(S_i)} \nabla_{\theta^\mu}\mu(S_i|\theta^\mu). \tag{8}$$

By combining continuous-control learning with OU noise-driven exploration and prioritized replay, the EDDPG framework produces a control policy that is not only efficient but fundamentally resilient to the threat of adversarial data manipulation. It also achieves faster convergence, reduced post-convergence variance, and improved obstacle avoidance capability compared to standard DDPG. Furthermore, the direct output of control commands eliminates the need for additional trajectory-to-actuation conversion layers, enabling integrated planning and control that is both computationally efficient and operationally robust in dynamic environments. The proposed EDDPG is summarized in Algorithm 1.

IV. Reward Function Design

The reward function is the central component of our defensive framework; it acts as the agent's security policy, teaching it how to behave in an adversarially compromised environment. Its design is critical for shaping a policy that is not only efficient but inherently resilient to deception. The primary challenge is to balance the core mission objective, reaching the target, with the critical security task of identifying and ignoring malicious sensory data. Thus, the reward is formulated as a weighted sum of physically interpretable terms:

$$R_t = w_1 r_{1t} + w_2 r_{2t} + w_3 r_{3t} + w_4 r_{4t} + w_5 r_{5t}, \tag{9}$$

where w_i denotes the weight of each reward component.

The design of R_t follows three principles: (1) provide dense and continuous feedback to accelerate convergence, (2) encode safety constraints directly into the reward to avoid unsafe exploration, and (3) balance short-term control stability with long-term goal achievement. Each component is derived from measurable quantities in the state vector, ensuring that the reward is grounded in observable and physically features.

Target Acquisition Reward: A large terminal reward encourages the agent to reach the target:

$$r_{1t} = \begin{cases} R_{\text{goal}}, & \text{if } \|(x_t, y_t) - (x_{\text{target}}, y_{\text{target}})\| \leq r_{\text{target}}, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where $R_{\text{goal}} \gg 0$.

Obstacle Avoidance Penalty: To discourage proximity to obstacles, a continuous penalty inversely proportional to the minimum distance is applied:

$$r_{2t} = \text{clip}\left(-\frac{1}{\min(D_{\text{obs}})}, r_{2\,\min}, 0\right), \tag{11}$$

where D_{obs} denotes the set of distances to the closest obstacles and $r_{2\,\min}$ bounds the penalty to avoid reward explosion.

This penalty is not applied to the adversarial obstacles since they do not actually exist. Doing this allows the RL agent to learn to differentiate between real and injected obstacles.

Progress Reward: Forward progress toward the target is incentivized via:

$$r_{3t} = -\kappa_3 \frac{\text{DistToTarget}_t}{\text{DistToTarget}_{t=0}}, \tag{12}$$

which is scale-invariant across different start-goal configurations.

Boundary Violation Penalty: Leaving the operational domain incurs a large negative penalty:

$$r_{4t} = \begin{cases} -R_{\text{bound}}, & \text{if out\_of\_bounds}, \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

Heading Alignment Reward: Efficient navigation is promoted by aligning the 's heading with the target direction:

$$r_{5t} = \kappa_5 \cos\left(2 \cdot \Delta h_t\right), \tag{14}$$

where Δh_t is the angular difference between the current heading and the desired heading toward the target.

V. Results and Discussion

This section presents a comprehensive evaluation of our proposed cyber-resilient EDDPG framework. The primary objective is to assess its effectiveness in maintaining mission integrity while under a simulated AOI attack. We compare its performance against baseline DDPG using Gaussian noise [27] and Artificial Potential Field (APF) navigation algorithm [28]. The analysis focuses on training stability, trajectory efficiency under attack, and overall mission success rates.

The simulation environment was configured as a 100×100 m populated with randomly placed static obstacles. For the attack scenarios, a single phantom obstacle was strategically injected into the AGV's perception data, continuously positioned along its optimal path to the target. RL algorithms were trained for $10,000$ episodes, each capped at 500 steps, across diverse obstacle layouts to ensure robustness. APF was executed in real time without pre-training.

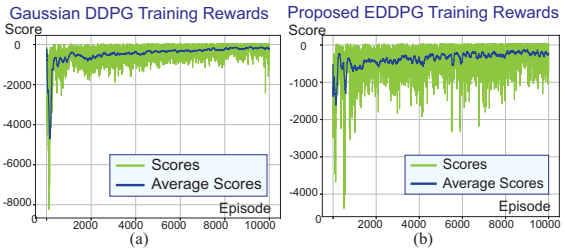

Fig. 3. Training performance of reinforcement learning algorithms. The progression of cumulative reward over 10,000 training episodes is shown for baseline DDPG with Gaussian noise, and the proposed EDDPG with OU noise. Gaussian DDPG converged but at a lower value than expected. The proposed EDDPG model successfully explored the environment and converged at a higher reward but with a higher variance. The higher variance is expected since the EDDPG agent was trained in an adversarial environment.

The reward evolution during training is shown in Fig. 3. baseline DDPG with Gaussian noise converged successfully, requiring about 10 hours of training. It achieved reliable navigation in most runs. The Agent converged to a lower reward than expected, which can be interpreted as unsuccessful complete exploration of the environment. It was also successfully deceived by the AOI attacks. Having never been exposed to malicious data during its training, it learned to treat all obstacles as legitimate threats. Consequently, when the malicious obstacle was injected, the agent diligently maneuvered to avoid it, resulting in a significantly diverted and suboptimal trajectory. This finding underscores a key insight: intelligence alone is insufficient for security. An agent is only as trustworthy as the data it is trained on.

Our adversarially trained EDDPG agent demonstrated remarkable resilience. Having learned to distinguish between real and malicious obstacle data via its security-aware reward function, the agent correctly identified the malicious obstacle as not a threat. It proceeded to largely ignore the injected data, maintaining a trajectory that was direct, efficient, and closely aligned with the optimal path. While minor deviations were sometimes observed as the agent evaluated the perceived threat, its ability to prioritize the true mission objective showcases the success of our adversarial training regimen.

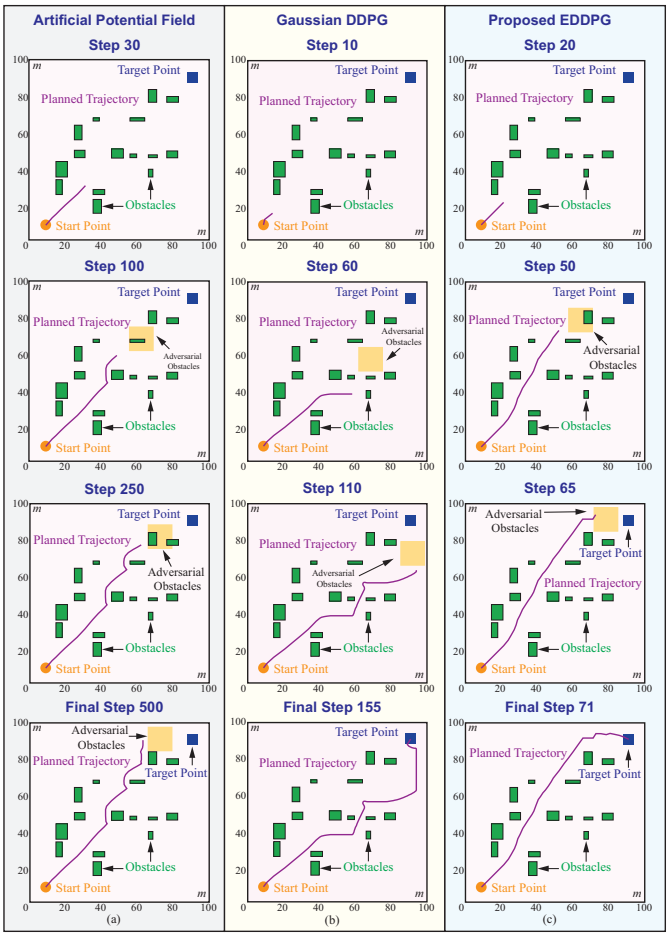

Fig. 4. Comparative trajectory results across different algorithms in an adversarial attack environment. Each column corresponds to a navigation method: APF, Gaussian DDPG, and the proposed EDDPG. Each row showcases the progress of the agent through the environment. Obstacles are shown as static rectangular blocks. Gaussian DDPG learned to avoid obstacles but was unable to distinguish between real and spoofed obstacles; APF became trapped in local minima behind the spoofed obstacle; and the proposed EDDPG achieved smooth, efficient, and collision-free trajectories ignoring the spoofed obstacle.

Traditional planners operate without pre-training, instead computing feasible paths online at each timestep. APF is prone to local minima and yielded geometric paths that require integration with a separate low-level controller to translate into throttle and steering commands. This distinction underscores the practical advantage of RL-based approaches, which directly output continuous control actions. The agent also proved extremely vulnerable to the AOI attack. As a purely reactive method that treats all perceived obstacles as repulsive forces, it had no mechanism to question the integrity of its sensory input. The malicious obstacles created a persistent repulsive field that easily diverted the AGV, often trapping it in local minima or forcing it onto extremely inefficient, circuitous paths. This demonstrates the inherent brittleness of traditional planners against deceptive data manipulation. Fig. 4 compares representative trajectories across all algorithms. Gaussian DDPG produced feasible but occasionally erratic paths and APF frequently became stuck in local minima. By contrast,

the proposed EDDPG consistently produced smooth, efficient, and collision-free trajectories, even under varying conditions.

Table I summarizes the estimated performance metrics. Both the traditional DDPG and traditional path planning algorithms were not successful in avoiding the obstacles while being resilient to adversarial attacks. The simulation studies highlight three insights: (i) RL methods significantly reduce online computation compared to traditional planners; (ii) Gaussian DDPG suffers from unstable policies; (iii) the proposed EDDPG achieves superior convergence stability, path efficiency, and adaptability to adversarial attack environments.

TABLE I
COMPARATIVE PERFORMANCE OF NAVIGATION ALGORITHMS

Algorithm	Success Rate	Avg. Steps to Target	Convergence Stability
APF	30%	High (>450 steps)	Sensitive to local minima
Gaussian DDPG	85%	Medium (∼150 steps)	Converged, No Adversarial Training
Proposed EDDPG	98%	Low (∼100 steps)	Stable, Adversarial Training

VI. CONCLUSION

This study addressed the vulnerability of autonomous ground vehicle navigation to adversarial obstacle injection by developing a cyber-resilient reinforcement learning framework. Through the integration of adversarial training within the proposed EDDPG algorithm, a navigation policy was obtained that remains effective even under deceptive sensing conditions. The findings confirm that resilience is not an emergent property of intelligence, but rather a feature that must be deliberately engineered into learning-based control systems. The proposed approach provides a practical pathway toward secure and trustworthy autonomous navigation, with future work directed toward extending resilience across multi-sensor fusion and real-world deployments.

REFERENCES

[1] J. H. Rogers III, T. Sellers, T. Lei, C. R. Hudson, and C. Luo, "Centroid-based cell decomposition robot path planning algorithm integrated with a bio-inspired approach," in *SPIE Conference: Unmanned Systems Technology XXVI*, vol. 13055. SPIE, 2024, pp. 43–51.

[2] T. Lei, P. Chintam, D. W. Carruth, G. E. Jan, and C. Luo, "Human-autonomy teaming-based robot informative path planning and mapping algorithms with tree search mechanism," in *2022 IEEE 3rd International Conference on Human-Machine Systems*. IEEE, 2022, pp. 1–6.

[3] T. Sellers, T. Lei, C. Luo, Z. Bi, and G. E. Jan, "Human autonomy teaming-based safety-aware navigation through bio-inspired and graph-based algorithms," *Biomimetic Intelligence and Robotics*, vol. 4, no. 4, p. 100189, 2024.

[4] T. Lei, C. Luo, S. X. Yang, D. W. Carruth, and Z. Bi, "Bio-inspired intelligence-based multi-agent navigation with safety-aware considerations," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 6, pp. 2946–2961, 2024.

[5] J. H. Rogers, T. Sellers, T. Lei, D. W. Carruth, and C. Luo, "Sensor-based multi-waypoint autonomous robot navigation with graph-based models," in *Autonomous Systems: Sensors, Processing and Security for Ground, Air, Sea, and Space Vehicles and Infrastructure 2023*, vol. 12540. SPIE, 2023, pp. 215–224.

[6] T. Lei, C. Luo, G. E. Jan, and Z. Bi, "Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm," *Frontiers in Robotics and AI*, vol. 9, 2022.

[7] T. Sellers, T. Lei, D. Carruth, and C. Luo, "Deep learning-based heterogeneous system for autonomous navigation," in *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping VIII*, vol. 12539. SPIE, 2023, pp. 140–153.

[8] T. Lei, P. Chintam, C. Luo, L. Liu, and G. E. Jan, "A convex optimization approach to multi-robot task allocation and path planning," *Sensors*, vol. 23, no. 11, p. 5103, 2023.

[9] B. Black, T. Sellers, T. Lei, C. Luo, and D. W. Carruth, "Optimal multi-target navigation via graph-based algorithms in complex environments," in *2024 IEEE 33rd International Symposium on Industrial Electronics (ISIE)*. IEEE, 2024, pp. 1–6.

[10] T. Lei, P. Chintam, C. Luo, and S. Rahimi, "Multi-robot directed coverage path planning in row-based environments," in *2022 IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. IEEE, 2022, pp. 114–121.

[11] E. Jayaraman, T. Lei, S. Rahimi, S. Cheng, and C. Luo, "Immune system algorithms to environmental exploration of robot navigation and mapping," in *Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12*. Springer, 2021, pp. 73–84.

[12] T. Lei, C. Luo, T. Sellers, Y. Wang, and L. Liu, "Multitask allocation framework with spatial dislocation collision avoidance for multiple aerial robots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 5129–5140, 2022.

[13] G. E. Jan, T. Lei, C.-C. Sun, Z.-Y. You, and C. Luo, "On the problems of drone formation and light shows," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 3, pp. 5259–5268, 2024.

[14] T. Lei, T. Sellers, C. Luo, D. W. Carruth, and Z. Bi, "Graph-based robot optimal path planning with bio-inspired algorithms," *Biomimetic Intelligence and Robotics*, p. 100119, 2023.

[15] T. Sellers, T. Lei, C. Luo, G. E. Jan, and J. Ma, "A node selection algorithm to graph-based multi-waypoint optimization navigation and mapping," *Intelligence & Robotics*, vol. 2, no. 4, pp. 333–54, 2022.

[16] T. Lei, C. Luo, T. Sellers, and S. Rahimi, "A bat-pigeon algorithm to crack detection-enabled autonomous vehicle navigation and mapping," *Intelligent Systems with Applications*, vol. 12, p. 200053, 2021.

[17] D. Short, T. Lei, D. W. Carruth, C. Luo, and Z. Bi, "A bio-inspired algorithm in image-based path planning and localization using visual features and maps," *Intelligence & Robotics*, vol. 3, no. 2, pp. 222–41, 2023.

[18] T. Lei, T. Sellers, C. Luo, L. Cao, and Z. Bi, "Digital twin-based multi-objective autonomous vehicle navigation approach as applied in infrastructure construction," *IET Cyber-Systems and Robotics*, vol. 6, no. 2, p. e12110, 2024.

[19] P. Chintam, T. Lei, B. Osmanoglu, Y. Wang, and C. Luo, "Informed sampling space driven robot informative path planning," *Robotics and Autonomous Systems*, p. 104656, 2024.

[20] T. Lei, C. Luo, G. E. Jan, and K. Fung, "Variable speed robot navigation by an ACO approach," in *Advances in Swarm Intelligence: 10th International Conference, ICSI 2019, Chiang Mai, Thailand, July 26–30, 2019, Proceedings, Part I 10*. Springer, 2019, pp. 232–242.

[21] J. Wang, F. Li, X. Zhang, and H. Sun, "Adversarial obstacle generation against lidar-based 3d object detection," *IEEE Transactions on Multimedia*, vol. 26, pp. 2686–2699, 2023.

[22] J. Banfi, Y. Zhang, G. E. Suh, A. C. Myers, and M. Campbell, "Path planning under malicious injections and removals of perceived obstacles: A probabilistic programming approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6884–6891, 2020.

[23] A. Szvoren, J. Liu, D. Kanoulas, and N. Tuptuk, "Exploring adversarial obstacle attacks in search-based path planning for autonomous mobile robots," *arXiv preprint arXiv:2504.06154*, 2025.

[24] T. Lei, T. Sellers, C. Luo, and L. Zhang, "A bio-inspired neural network approach to robot navigation and mapping with nature-inspired algorithms," in *Advances in Swarm Intelligence: 13th International Conference, ICSI 2022, Xi'an, China, July 15–19, 2022, Proceedings, Part II*. Springer, 2022, pp. 3–16.

[25] Z. Chu, F. Wang, T. Lei, and C. Luo, "Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 108–120, 2022.

[26] N. Yan, S. Huang, and C. Kong, "Reinforcement learning-based autonomous navigation and obstacle avoidance for USVs under partially observable conditions," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 5519033, 2021.

[27] D. Wang and M. Hu, "Deep deterministic policy gradient with compatible critic network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 4332–4344, 2021.

[28] S. Alarabi, T. Lei, M. Santora, C. Luo, and T. Sellers, "Multi-robot path planning using potential field-based simulated annealing approach," in *Unmanned Systems Technology XXVI*, vol. 13055. SPIE, 2024, pp. 102–117.