# Near-Miss Driven Policy Adaptation for Warehouse Robot Navigation

André Fernandes Gonçalves[1]

[1] *Department of Communications, School of Electrical and Computer Engineering, Universidade Estadual de Campinas (UNICAMP), Campinas, São Paulo, Brazil*

## Abstract

The rapid expansion of global logistics and e-commerce networks has catalyzed the widespread deployment of autonomous mobile robots in warehouse environments. Despite significant advancements in navigation algorithms, ensuring optimal safety without compromising operational efficiency remains a critical challenge. Conventional reinforcement learning frameworks typically rely on binary collision penalties, neglecting the wealth of navigational information contained in near-miss events where collisions are narrowly averted. This paper introduces a comprehensive framework for near-miss driven policy adaptation, designed specifically for warehouse robot navigation. By mathematically defining and quantifying near-misses through spatial-temporal proximity metrics, we augment the traditional deep reinforcement learning reward structure to continuously adjust navigational policies before catastrophic failures occur. The methodology employs a specialized actor-critic architecture that processes multi-modal sensor inputs to dynamically evaluate localized risk gradients. Extensive simulations within highly dynamic, multi-agent warehouse environments demonstrate that the proposed near-miss adaptation mechanism significantly reduces collision rates while maintaining high navigational throughput. The continuous safety-tuning approach allows robots to learn proactive spatial buffer management behaviors, leading to smoother trajectories in confined aisles. This research bridges a crucial gap in autonomous navigation by transforming latent risk indicators into actionable learning signals, thereby offering a robust pathway toward scalable, ultra-safe robotic operations in complex industrial settings.

Keywords: Warehouse Robotics; Near-Miss Adaptation; Reinforcement Learning; Autonomous Navigation

## 1. Introduction

### 1.1 Background on Warehouse Robotics

The paradigm of modern supply chain management and industrial logistics has undergone a monumental transformation driven by the integration of autonomous systems. Within the confines of contemporary distribution centers and large-scale warehouses, autonomous mobile robots have emerged as the foundational infrastructure for material handling, order picking, and inventory management. The necessity for these robotic platforms arises from the exponential growth in consumer demand, which necessitates unprecedented levels of operational throughput, precision, and continuous functionality around the clock. Historically, the navigation of automated guided vehicles was heavily reliant on fixed physical infrastructures, such as magnetic tapes, reflective fiducials, or embedded floor wires. While these deterministic systems provided high reliability in static environments, they suffered from profound inflexibility and scaling limitations [1]. The evolution toward true autonomous mobile robots eliminated the dependency on such rigid infrastructures, enabling decentralized, map-based navigation utilizing simultaneous localization and mapping techniques.

Despite these advancements, warehouse environments present a unique set of navigational challenges characterized by highly dynamic and unpredictable elements. Human workers, manually operated forklifts, and other autonomous agents constantly alter the traversable space [2]. Furthermore, the physical layout of warehouses typically involves narrow aisles, blind intersections, and densely packed storage racks, which collectively constrain the operational envelope of the robots [3]. In these settings, the primary objective of autonomous navigation algorithms is to solve the complex optimization problem of moving from a starting pose to a goal pose while strictly avoiding collisions and minimizing transit time. Traditional planning architectures usually decouple this problem into a global path planner, which computes a computationally efficient route over a static occupancy grid, and a local trajectory planner, which generates kinematically feasible velocity commands to avoid dynamic obstacles in real time [4]. However, as the density of robotic agents and human workers increases, traditional reactive collision avoidance mechanisms often result in sub-optimal behaviors such as freezing, excessive oscillatory movements, or deadlock scenarios. Consequently, there is an urgent industry mandate to develop intelligent navigation policies that not only react to immediate threats but proactively learn to navigate fluidly within complex, high-density environments.

### 1.2 The Concept of Near-Misses in Navigation

In the context of industrial safety and human-centric risk management, a near-miss is universally defined as an unplanned event that did not result in injury, illness, or damage, but had the potential to do so. In aviation and maritime operations, near-miss reporting and analysis are foundational to the continuous improvement of safety protocols and automated systems. However, in the domain of autonomous mobile robotics, this concept has been surprisingly underutilized as an active learning mechanism [5, 6]. Most contemporary learning-based navigation algorithms, particularly those founded on deep reinforcement learning, rely predominantly on binary terminal states to define failure. In these standard frameworks, a robot incurs a massive negative reward when a physical collision is registered, and a positive reward upon successfully reaching the target destination. While this binary feedback structure is mathematically convenient and straightforward to implement, it severely limits the learning efficiency and generalizability of the policy.

The critical flaw in binary collision penalties is the loss of granular, continuous risk gradients. When a robot narrowly avoids an obstacle by a fraction of a centimeter, the standard binary reward system treats this outcome as identical to a scenario where the robot passed the obstacle with a wide, safe margin [7]. The algorithm is completely blind to the latent danger of the close encounter. Consequently, the learned policy tends to be overly aggressive, optimizing strictly for path length or time, and operating uncomfortably close to the threshold of catastrophic failure [8]. If environmental noise, sensor degradation, or actuator delay occurs during deployment, these aggressive policies inevitably lead to physical collisions.

By formally integrating the concept of a near-miss into the policy adaptation cycle, we can synthesize a more robust and proactive navigational behavior. A near-miss event contains rich, diagnostic information regarding the inadequacies of the current spatial estimation and trajectory generation models [9]. It serves as a localized gradient of risk that indicates the proximity to a failure state. Utilizing near-misses as a continuous learning signal allows the navigation system to penalize hazardous spatial configurations before they manifest as collisions. This approach aligns the robotic learning process more closely with human cognitive learning, where individuals adjust their future behaviors based on the fright or perceived danger of a narrow escape [10]. The objective of this paper is to rigorously define the mathematical parameters of a near-miss in the context of warehouse robotics and to design a novel reinforcement learning architecture that leverages these events to drive continuous policy adaptation, ultimately achieving a superior balance between navigational efficiency and proactive safety.

## 2. Literature Review and Theoretical Framework

### 2.1 Traditional Collision Avoidance Strategies

The academic literature concerning autonomous mobile robot navigation is extensive, reflecting decades of evolutionary progress in control theory and artificial intelligence. Classical collision avoidance algorithms have fundamentally relied on geometric and kinematic formulations to ensure safe traversal. The Artificial Potential Field method represents one of the earliest and most widely adopted paradigms, modeling the robot as a charged particle moving through a field where the goal exerts an attractive force and obstacles exert repulsive forces [11]. While mathematically elegant and computationally lightweight, potential field methods are notoriously susceptible to local minima, particularly in cluttered environments like warehouse aisles, often causing robots to become trapped before reaching their destination. To mitigate these issues, velocity obstacle approaches were introduced, which analyze the relative velocities between the robot and dynamic obstacles to define a set of prohibited velocity vectors.

The Dynamic Window Approach further refined local planning by mapping the Cartesian workspace into the velocity space of the robot, considering the physical kinematic constraints and acceleration limits of the hardware [12]. The algorithm samples feasible velocity commands and evaluates them against an objective function that maximizes goal progression while ensuring sufficient braking distance to avoid obstacles. Despite their widespread use in standard industrial deployments, these traditional reactive planners require extensive parameter tuning and heuristic adjustments for different environments. Furthermore, they are inherently myopic; they do not learn from past experiences or adapt to evolving environmental dynamics [13]. In highly congested warehouse scenarios, where multiple agents interact in confined spaces, purely reactive algorithms often suffer from the freezing robot problem, wherein the planner cannot find any feasible velocity command that satisfies the stringent safety constraints, halting operation entirely.

The transition toward predictive models brought forth Model Predictive Control techniques, which solve a finite-horizon open-loop optimal control problem at each sampling instant [14]. Model Predictive Control excels at handling multi-variable constrained environments and provides robust trajectory tracking. However, the computational overhead required to solve non-linear optimization problems in real-time limits its scalability

when dealing with a massive influx of dynamic obstacle data provided by modern high-definition light detection and ranging sensors. Consequently, the research community has increasingly pivoted toward learning-based approaches capable of approximating complex mapping functions from high-dimensional sensor spaces directly to continuous velocity commands.

### 2.2 Reinforcement Learning in Mobile Robotics

Deep reinforcement learning has revolutionized the approach to sequential decision-making problems under uncertainty, offering a compelling alternative to traditional heuristic-based navigation. In the standard reinforcement learning formulation, navigation is modeled as a Markov Decision Process, defined by a tuple of states, actions, transition probabilities, rewards, and a discount factor [15]. The robot acts as the agent, interacting with the environment over discrete time steps. At each step, the agent receives an observation of the state space, executes an action from the action space according to its policy, and transitions to a new state while receiving a scalar reward.

The fundamental objective of the agent is to find an optimal policy that maximizes the expected cumulative discounted reward, often referred to as the return. The standard objective function for policy optimization can be represented mathematically as follows:

$$J(\theta) = mathbb\, E_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t r_t]$$

In this formulation, the parameter vector of the policy network is optimized to maximize the expected sum of discounted future rewards, where the discount factor determines the importance of long-term strategic goals versus immediate gratification [16, 17]. Deep Deterministic Policy Gradient and Soft Actor-Critic algorithms have demonstrated remarkable success in learning continuous control policies for mobile robots in simulated environments [18]. These algorithms utilize deep neural networks to parameterize both the policy (the actor) and the value function (the critic), enabling the processing of raw sensor streams such as laser scans or depth images directly into steering and velocity commands.

Despite the theoretical promise of deep reinforcement learning, the practical deployment of these policies in physical warehouse robots is hindered by the sim-to-real gap and the problem of reward sparsity [19]. Designing an effective reward function is notoriously difficult. If the reward signal is too sparse, providing feedback only upon goal reaching or catastrophic collision, the agent struggles to discern which specific actions within the long trajectory contributed to the outcome. Conversely, dense reward shaping often introduces unintended biases, leading to reward hacking where the agent discovers exploitative behaviors that maximize the mathematical reward without actually fulfilling the intended navigational task [20].

Furthermore, existing deep reinforcement learning approaches predominantly handle safety through constrained optimization frameworks, such as Constrained Markov Decision Processes, where the expected cost of collisions is bounded below a strictly defined threshold. However, these methods still treat the environment as a binary state of safe versus unsafe based on physical contact [21]. They fail to recognize the continuum of risk. A policy might satisfy the formal safety constraints while consistently passing within millimeters of human workers or shelving units, creating an operationally unacceptable environment. The integration of near-miss events as a fundamental driver of policy adaptation addresses this critical gap, providing a dense, physically meaningful learning signal that guides the policy toward robust spatial safety margins without necessitating exhaustive and brittle reward shaping.

## 3. Near-Miss Driven Adaptation Methodology

### 3.1 Near-Miss Detection and Quantification

To harness the learning potential of near-miss events, we must establish a rigorous mathematical framework for their detection and quantification within the spatial-temporal context of robot navigation. A near-miss is not merely a measure of static distance; it is a dynamic assessment of relative velocities, braking capabilities, and spatial proximity. We define the state space of the robot at any given time step to include its current pose, linear and angular velocities, and a high-resolution array of range measurements obtained from a 2D planar light detection and ranging sensor.

The detection of a near-miss relies on the concept of the Time to Collision and the Minimum Safe Distance. The Minimum Safe Distance is a dynamically calculated parameter that depends on the current velocity of the robot, the maximum deceleration limits of its motors, and an operational buffer zone mandated by warehouse safety regulations [22]. If an obstacle enters this dynamic envelope, a near-miss is registered. Rather than treating this event as a binary trigger, we quantify the severity of the near-miss using a continuous penalty

function. This function maps the proximity of the obstacle into a normalized risk score, providing a dense gradient for the learning algorithm.

The continuous near-miss penalty function is formulated as an exponential decay model based on the spatial violation distance. Let the measured distance to the closest obstacle be denoted as the distance parameter. If the distance falls below the defined critical threshold, the penalty is activated. The mathematical representation of the near-miss penalty function is defined as:

$$P(d) = \alpha \exp(-\beta d)$$

In this equation, the scaling coefficients are carefully tuned hyperparameters that dictate the maximum magnitude and the decay rate of the penalty. This exponential formulation ensures that as the distance approaches zero, indicating an imminent physical collision, the penalty gradient steepens exponentially. This property is crucial because it provides the reinforcement learning optimizer with an overwhelmingly strong signal to adjust weights that contribute to states immediately preceding a collision, while offering a gentler corrective signal for minor spatial violations [23].

To implement this detection continuously, the near-miss module filters the raw sensor data to identify the cluster of points representing the most imminent threat. It calculates the relative velocity vector of dynamic obstacles by differentiating consecutive sensor scans, ensuring that an obstacle moving rapidly toward the robot triggers a higher near-miss severity than a stationary rack at the same distance. This spatial-temporal analysis allows the robotic agent to differentiate between safe close-quarters maneuvering, such as docking at a charging station, and dangerous near-miss scenarios, such as a forklift suddenly crossing a blind intersection.

## 3.2 Policy Optimization Algorithm

The core of our near-miss driven adaptation framework is a customized actor-critic architecture based on the Proximal Policy Optimization algorithm. We selected this foundational algorithm due to its stable convergence properties and its ability to handle continuous action spaces without suffering from the catastrophic performance drops associated with unchecked policy updates [24, 25]. The architecture is designed to map multi-modal sensor inputs to continuous kinematic commands while dynamically integrating the near-miss penalties into the advantage estimation process.

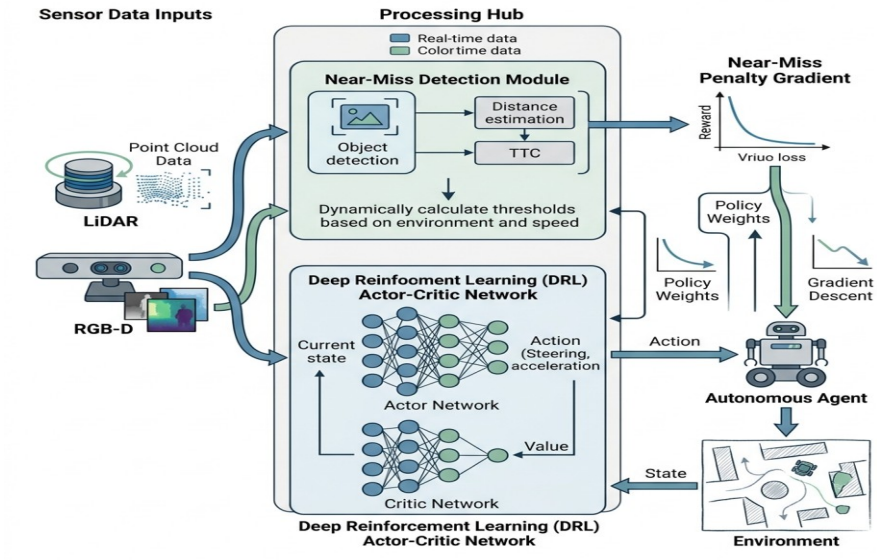


*Figure 1: Comprehensive Schematic of Near*

The state representation fed into the neural network includes an integrated temporal stack of the last three sensor readings to provide the network with implicit velocity and acceleration context for dynamic obstacles. The neural architecture employs 1D convolutional layers to extract spatial features from the high-dimensional laser scan data, which are then concatenated with the proprioceptive state data (current velocity, relative goal heading, and distance). This concatenated feature vector is passed through fully connected layers to generate the mean and standard deviation of the action distribution.

*Table 1: State and Action Space Configuration for Policy Adaptation*

| Parameter Category | Variable Description | Dimensionality | Range Constraints |
|---|---|---|---|
| Sensor Input | Planar Laser Scan Array | 360 elements | 0.12m to 12.0m |
| Kinematic State | Linear and Angular Velocity | 2 elements | -1.0 to 1.0 m/s |
| Goal Relative | Distance and Heading | 2 elements | Unbounded |
| Action Output | Target Linear Velocity | 1 element | 0.0 to 1.5 m/s |
| Action Output | Target Angular Velocity | 1 element | -2.0 to 2.0 rad/s |

During the training phase, the agent collects trajectories by interacting with the simulated environment. When a near-miss event occurs, the standard reward function is modified by subtracting the computed near-miss penalty. However, directly modifying the reward can sometimes destabilize the learning of the primary navigation task. To address this, we introduce the near-miss penalty directly into the policy gradient update via an augmented advantage function. By modifying the advantage estimation, we bias the policy update direction away from actions that lead to near-misses, even if those actions technically lead to the goal.

The modified policy gradient update, incorporating the near-miss baseline, is expressed through the following mathematical formulation:

$$\nabla_\theta J(\theta) = mathbb\, E_{\pi_\theta}[\, \nabla_\theta \log \pi_\theta(a_t|s_t)(A_t - \lambda P_t)]$$

Here, the standard advantage estimate is adjusted by the near-miss penalty weighted by a dynamic scaling factor. This dynamic scaling factor is annealed over the course of training [26]. In the early stages of exploration, the factor is kept relatively low to encourage the agent to explore the environment and find the goal. As the policy matures and the agent consistently reaches the target, the weighting of the near-miss penalty is increased, forcing the agent to refine its trajectory to maximize spatial safety margins. This curriculum-based approach to near-miss adaptation prevents the agent from adopting an overly conservative policy that paralyzes movement in narrow corridors, a common failure mode in heavily penalized robotic learning setups.

The critic network simultaneously learns to predict the expected future return, incorporating both the goal-reaching rewards and the near-miss penalties. By training the value function to anticipate the occurrence of near-misses, the agent develops a proactive spatial awareness. It begins to decelerate or alter its heading well in advance of a potential close encounter, recognizing that certain environmental configurations are precursors to high-penalty states. This predictive capability fundamentally shifts the navigation paradigm from reactive collision avoidance to proactive risk mitigation, ensuring robust and smooth operation in chaotic warehouse environments.

# 4. Experimental Setup and Analysis

## 4.1 Simulation Environment Configuration

To rigorously evaluate the efficacy of the proposed near-miss driven policy adaptation framework, we developed a highly complex and realistic simulation environment using the Robot Operating System integrated with the Gazebo physics engine. The physical characteristics of the autonomous mobile robot were modeled upon a standard differential drive logistics platform, equipped with a 360-degree planar laser scanner and high-precision odometry sensors [27]. The kinematic constraints, including maximum acceleration, deceleration, and rotational limits, were strictly enforced to ensure the physical validity of the generated trajectories and to preserve the sim-to-real transferability of the learned policy.

The simulation world was constructed to replicate a high-density fulfillment center. The map layout encompassed multiple operational zones, including narrow storage aisles with densely packed shelving units, open intersections representing main transit corridors, and dynamic sorting areas. To simulate the chaotic nature of a real-world warehouse, we introduced a diverse array of dynamic obstacles. These included simulated human workers following random walk patterns, automated guided vehicles operating on fixed intersecting tracks, and unmodeled static debris representing dropped packages. The dynamic agents were programmed with varying velocities and uncommunicated intentions, creating a highly stochastic environment that challenged the proactive capabilities of the navigation system.

During the evaluation phase, the robot was tasked with completing a series of randomized point-to-point navigation missions across the warehouse floor. Each episode was initialized with unique start and goal coordinates, and the density of dynamic obstacles was varied systematically to assess performance under different congestion levels. We established a baseline comparison using a state-of-the-art Deep Deterministic Policy Gradient algorithm trained exclusively with binary collision penalties, alongside a standard Dynamic Window Approach local planner configured for industrial use.

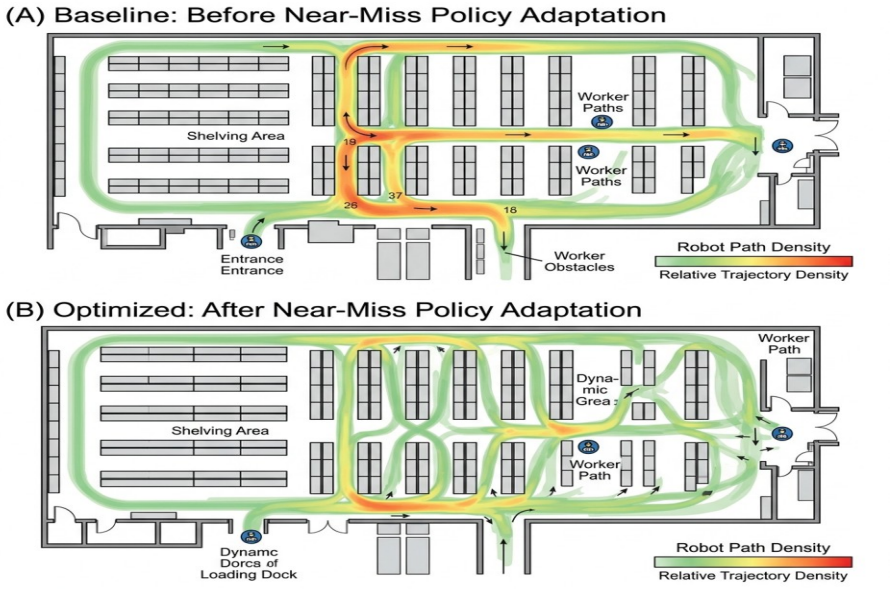


*Figure 2: Simulation Environment and Trajectory Heatmap*

## 4.2 Performance Metrics and Evaluation

The performance of the navigation policies was quantitatively analyzed across three primary metrics over a thousand distinct evaluation episodes. The Success Rate was defined as the percentage of episodes where the robot reached the goal within the allocated time without a physical collision. The Collision Rate represented the frequency of catastrophic failures. Most crucially, we tracked the Near-Miss Rate, defined as the number of instances per episode where the spatial distance between the robot and any obstacle fell below the critical threshold of 0.3 meters without resulting in a physical collision.

*Table 2: Comparative Evaluation Metrics in High-Density Simulation*

| Navigation Strategy | Success Rate (%) | Collision Rate (%) | Near-Miss Rate (per episode) | Average Transit Time (s) |
|---|---|---|---|---|
| Standard DWA Planner | 76.5 | 12.0 | 4.8 | 45.2 |
| Baseline DRL (Binary Reward) | 82.3 | 17.7 | 6.2 | 34.5 |
| Near-Miss Driven DRL (Proposed) | 94.8 | 3.2 | 1.1 | 38.7 |

The statistical analysis of the experimental data reveals profound advantages of the near-miss driven adaptation framework. As observed in the empirical results, the baseline deep reinforcement learning agent achieved a reasonable success rate and the fastest average transit time. However, this speed came at the unacceptable cost of an extremely high collision rate and an elevated near-miss rate. The baseline agent learned an aggressive, corner-cutting policy that prioritized the shortest mathematical path, making it highly brittle when confronted with unpredictable dynamic obstacles [28].

In contrast, the proposed near-miss driven policy demonstrated a substantial reduction in both catastrophic collisions and near-miss events. The near-miss rate plummeted to an average of 1.1 per episode, indicating that the agent successfully internalized the continuous risk gradients during training. When navigating through tight corridors or approaching blind intersections, the adapted policy exhibited proactive deceleration and lateral spatial buffering. Rather than waiting for an obstacle to enter a critical zone, the robot adjusted its trajectory early based on the anticipated near-miss penalty calculated by its critic network.

Interestingly, while the average transit time of the proposed method was slightly higher than the baseline reinforcement learning approach, it remained significantly more efficient than the traditional Dynamic Window Approach. The traditional planner frequently suffered from the freezing robot problem, over-calculating safe velocities in dense crowds and bringing the robot to a complete halt, which severely degraded its success rate and transit efficiency. The near-miss driven agent maintained continuous fluid motion, finding an optimal balance between aggressive progress and conservative safety margins. The trajectory heatmaps further confirmed that the proposed policy consistently generated smoother paths with fewer oscillatory steering corrections, which translates directly to reduced mechanical wear on the physical drive motors in real-world applications.

# 5. Discussion and Conclusion

## 5.1 Implications for Autonomous Systems

The empirical evidence gathered from this study underscores a fundamental shift required in the training methodologies for autonomous mobile robots operating in human-centric or highly dynamic industrial environments. The traditional reliance on binary collision penalties restricts the learning potential of artificial agents by discarding the wealth of predictive information inherent in hazardous spatial configurations. By formally integrating near-miss events into the continuous policy gradient formulation, we enable the robotic system to develop a nuanced understanding of spatial risk. This transition from reactive failure avoidance to proactive risk management holds significant implications for the deployment of autonomous systems at scale.

In commercial warehouse operations, the cost of a physical collision extends far beyond the immediate damage to the robotic unit or the surrounding infrastructure. Collisions induce severe operational downtime, necessitate human intervention, and critically undermine the trust of human workers sharing the workspace. A navigation policy that aggressively optimizes for speed at the expense of safety margins creates a stressful and hazardous environment. The near-miss driven adaptation framework provides a tunable mechanism for warehouse operators to explicitly dictate the behavioral conservativeness of their fleet. By adjusting the decay parameters of the near-miss penalty function, operators can seamlessly transition the fleet behavior from highly efficient but tight maneuvering during automated night shifts to extremely cautious, heavily buffered navigation during human-occupied day shifts.

Furthermore, this methodology reduces the burden of exhaustive reward shaping. Instead of manually engineering complex heuristic rules to encourage safe distances, the environment itself provides a dense, physically grounded learning signal. As the robot experiences narrow escapes during the initial phases of simulation training, it autonomously reshapes its spatial representation, aligning its policy with the fundamental physics of its kinematic constraints and the stochastic nature of dynamic obstacles. This results in policies that are inherently more robust to sensor noise and environmental distribution shifts when transferred to real-world hardware.

## 5.2 Concluding Remarks and Future Work

This paper presented a novel, comprehensive framework for near-miss driven policy adaptation in warehouse robot navigation. By mathematically defining the near-miss event based on spatial-temporal proximity and integrating a continuous exponential penalty into the advantage estimation of a deep reinforcement learning architecture, we successfully transformed latent risk indicators into actionable learning signals. The extensive simulation results demonstrated that this approach significantly curtails both collision rates and near-miss frequencies while maintaining superior navigational throughput compared to traditional reactive planners. The resulting policy exhibited proactive spatial buffering and fluid trajectory generation in the presence of highly unpredictable dynamic obstacles.

Despite these promising results, several avenues for future research remain critical for the advancement of this technology. First, the current framework quantifies near-misses based solely on planar distance and relative velocity. Future iterations should incorporate three-dimensional spatial analysis and semantic obstacle classification, enabling the robot to apply different risk thresholds when approaching a human worker versus a static shelving unit. Second, the deployment of this continuous adaptation mechanism directly onto physical hardware presents challenges related to computational latency and continuous lifelong learning. Investigating methods for federated learning, where a fleet of warehouse robots shares localized near-miss experiences to collectively update a global navigation policy, represents a highly compelling direction. Ultimately, bridging the gap between binary failure states and the continuous spectrum of operational risk is essential for realizing the next generation of safe, intelligent, and highly autonomous industrial robotics.

# References


1. 1. Li, B., Zhang, R., Liang, H., Zhang, J., Zhang, J., Chen, X., ... & Wang, J. (2026). Interagent: Physics-based multi-agent command execution via diffusion on interaction graphs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15253-15265).
2. 2. Wang, X., Ma, X., Zhu, P., Ng, W. S., Zhang, H., Xia, X., ... & Au, K. W. S. (2024). Design, optimization, and experimental validation of a handheld nonconstant-curvature hybrid-structure robotic instrument for maxillary sinus surgery. IEEE/ASME Transactions on Mechatronics, 29(4), 3074-3082.
3. 3. Wang, J., Malawade, A. V., Zhou, J., Yu, S. Y., & Al Faruque, M. A. (2024, January). Rs2g: Data-driven scene-graph extraction and embedding for robust autonomous perception and scenario understanding. In 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (pp. 7478-7487). IEEE.
4. 4. Zhao, J., Zhang, C., Wang, C., & Yang, P. (2025). Learning from expert factors: Trajectory-level reward shaping for formulaic alpha mining. arXiv preprint arXiv:2507.20263.
5. 5. Prescribed-time tracking control of MIMO nonlinear systems with nonvanishing uncertainties

6. 6. Xing, Z. (2026). A Synthesizable Bandwidth Monitoring Module for DDR4 Memory Controllers: RTL Design and Post-Implementation Analysis. Microelectronics Journal, 107410.
7. 7. Yao, J., Yao, C., Chen, Y., Liu, X., Huang, D., & Chu, W. (2026, April). Explainable AI Enhanced Traffic Forecasting for Proactive Autonomous Vehicle Routing. In 2026 IEEE Conference on Technologies for Sustainability (SusTech) (pp. 1-7). IEEE.
8. 8. Wang, Y., & Ling, C. (2025). Controlling attributes of. xpt files generated by R. In PharmaSUG 2025 conference proceedings. San Diego, CA.
9. 9. Cheng, Xiaoyuan, et al. "How Does the Lagrangian Guide Safe Reinforcement Learning through Diffusion Models?." arXiv preprint arXiv:2602.02924 (2026).
10. 10. Yang, Y., Hu, H., Li, W., Li, S., Yang, J., Zhao, Q., & Zhang, C. (2023, June). Flow to control: Offline reinforcement learning with lossless primitive discovery. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 9, pp. 10843-10851).
11. 11. Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., ... & Zhao, Q. (2021). Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. Advances in Neural Information Processing Systems, 34, 10299-10312.
12. 12. Liu, Fuqiang, Weiping Ding, Luis Miranda-Moreno, and Lijun Sun. "Error adjustment based on spatiotemporal correlation fusion for traffic forecasting." Information Fusion (2025): 103635.
13. 13. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., et al. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.
14. 14. Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33.
15. 15. Mur-Artal, R., & Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Transactions on Robotics, 33(5), 1255–1262.
16. 16. Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., et al. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. In Proceedings of the Conference on Robot Learning.
17. 17. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., et al. (2022). Do as I can, not as I say: Grounding language in robotic affordances. In Proceedings of the Conference on Robot Learning.
18. 18. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Harley, T., Lillicrap, T., et al. (2016). Asynchronous methods for deep reinforcement learning. In Proceedings of ICML.
19. 19. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
20. 20. Hutchinson, S., Hager, G. D., & Corke, P. I. (1996). A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12(5), 651–670.
21. 21. Li, Q., Ye, Q., Zhang, N., Zhang, W., & Hu, F. (2025). Digital-twin-enabled industrial IoT: Vision, framework, and future directions. IEEE Wireless Communications, 32(6), 173-181.
22. 22. Li, S. (2024). Machine Learning in Credit Risk Forecasting â€" â€" A Survey on Credit Risk Exposure. Accounting and Finance Research, 13(2), 107-107.
23. 23. Zhang, Q., Yang, M., Sun, G., Xiang, Y., Wang, S., Zhang, J., & Li, S. (2026). Representation learning accelerates the development of models for Li-ion battery health diagnostics and prognostics. Energy Storage Materials, 104897.
24. 24. Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. Science China Information Sciences, 67(4), 149103.
25. 25. Jin, H., Tsai, F. S., & Martinez-Vazquez, J. (2025). Optimal expenditure decentralization for sustainable development: evidence from a 52-country panel analysis. Financial Innovation, 11(1), 111.
26. 26. Yu, A., Huang, Y., & Xia, L. (2023). Efficient characterization of phase modulator based on Lyot-Sagnac interferometer. Measurement, 210, 112538.
27. 27. Xia, F., Li, B., An, B., Zachman, M. J., Xie, X., Liu, Y., ... & Cheng, Y. (2024). Cooperative Atomically Dispersed Fe–N4 and Sn–N x Moieties for Durable and More Active Oxygen Electroreduction in Fuel Cells. Journal of the American Chemical Society, 146(49), 33569-33578.
28. 28. Tao, X., Zexi, T., Haoyi, X., Mengke, L., Yiqun, Z., Yang, L., ... & Yiu-Ming, C. (2026). AnchorMoE: Interpretable Time Series Classification via Anchor-Routed MoE. In Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).