Research Article

Interpretable reinforcement learning framework for autonomous warehouse robot navigation

Suhail Ashfaq Butt[1], Umer Farooq[2], Muhammad Shan[3], Muhammad Faisal[4], Asghar Ali Shah[5], Omar Almomani[6], Taher M. Ghazal[6,7,8]

[1]Department of Computer Science, University of Education, Lahore 54000, Pakistan.
[2]Department of Computing, Faculty of Engineering Sciences and Technology, Hamdard University Karachi 74600, Pakistan.
[3]Department of Computer Science, Institute for Art and Culture, Lahore 54000, Pakistan.
[4]Department of Computer Science, Bahria University Lahore Campus, Lahore 54000, Pakistan.
[5]Department of Computer Science, Kateb University, Kabul 1001, Afghanistan.
[6]Department of Networks and Cybersecurity, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan.
[7]Faculty of Computing and IT, Sohar University, Sohar 311Oman.
[8]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Selangor, Malaysia.

Correspondence to: Dr. Asghar Ali Shah, Department of Computer Science, Kateb University, Kabul 1001, Afghanistan. Email: sghar.ali.shah@kateb.edu.af

Received: 16 May 2026 | Approved: 27 May 2026 | Online: 27 May 2026

Abstract

The dynamic nature of autonomous warehouses presents a host of challenges, including efficient path planning and obstacle avoidance, as well as the ability to handle tasks within the warehouse. Reinforcement learning (RL) has been effective in training an adaptive navigation policy but presents some risks in terms of safety, transparency, and trust because of its black-box nature. In this paper, we propose an interpretable reinforcement learning framework that aims for better performance and explainability in robot navigation for warehouse applications. The framework integrates deep reinforcement learning (DRL) and explainability methods like saliency mapping and feature importance analysis to enable humans to understand the process of decision-making. The framework combines these techniques to enable the robot's learned policies to be both efficient and interpretable, which facilitates understanding and trust of the robot's actions by the warehouse operators. The experimental results indicate that the "collision" outcome reached the maximum collision rate of 98%, and the average time and path smoothness were 31.87 and 0.539 respectively, demonstrating the effectiveness of the framework in terms of improving the performance and interpretability. The task completion times of the robot were decreased, and faster navigation was seen in successful runs as compared to the runs that resulted in timeout or collision. In addition, interpretability features like saliency maps and feature attribution analysis were used to build trust from the operators, as they clearly explained the robot's decision-making process. The framework helps to ensure the efficiency and reliability of warehouse robots, which could be used in various applications related to logistics, inventory management, and material handling.

Keywords: Interpretable reinforcement learning, autonomous warehouse robots, deep reinforcement learning, simulated navigation, explainability methods

INTRODUCTION

Background on autonomous warehouse robots

With the continuous growth of e-commerce, the warehouse logistics system is becoming more and more efficient and flexible, so the use of autonomous robots is increasingly important. The robots can be applied to different applications such as material handling and storage, order picking, sorting and inventory management. With the advent of autonomous technologies, these robots are able to navigate complex environments, interact with objects, and execute tasks with minimal human intervention. But the warehouse conditions are extremely challenging and conditions are constantly changing; there are obstacles, humans and other machines.

The robots have to navigate this environment effectively in order to be able to reliably and safely complete tasks. The systems are required to be able to plan adaptive path, navigate around obstacles and perform a series of tasks within a real time limit. Modern warehouses are complex environments, and robots need to adapt swiftly and adeptly when unexpected changes occur. To reach such a degree of adaptability, a lot of autonomous robots use machine learning (ML), and more specifically reinforcement learning (RL), which allows them to learn optimal behaviors through trial and error.

Even though these RL systems are promising, there are numerous issues with transparency and safety in current RL-based systems in robotics. RL algorithms learn from their experience, and develop complex decision making processes that are often hard to interpret for humans. This “black-box” character of RL presents tremendous challenges in critical environments such as a warehouse, where safety, trust and reliability matter.

Role of reinforcement learning (RL) in autonomous navigation

A type of machine learning called reinforcement learning is very successful at learning complex behaviors through interaction with the environment in the absence of a human teacher. Reinforcement learning is a class of machine learning algorithms that have been very successful at training an autonomous robot to learn a complex task without a human teacher. The RL algorithms learn to optimally perform a specific task by rewarding or punishing the robot based on its actions, which in turn allows the algorithm to gradually learn optimal strategies for the task. RL has potential applications in autonomous warehouse robots such as path planning, obstacle avoidance, and efficient task execution. The capacity to adapt to changing circumstances in the warehouse enables robots to make on-the-fly decisions for effective execution of tasks.

RL has proven useful in robots navigating in dynamic environments. However, traditional RL models are black-box models, which pose a challenge in safety-critical applications like navigation in a warehouse. The efficiency or success rate of RL has been used as the standard for measuring the performance of these algorithms, but the decision-making process by which they achieved efficiency or success is vague. This opacity renders the human operator's trust in the system's decisions uncertain and leads to safety concerns and reluctance to implement such systems in real-world applications.

The need for interpretability in RL for autonomous systems

Interpretability in Machine Learning is the ability to understand and explain how the model makes its decision. For autonomous robots, interpretability is crucial to establishing trust between the robot and its human operators. Warehouse workers need to be able to comprehend why the robot is doing what it's doing, particularly when safety is an issue. For example, if a robot unexpectedly alters its route or behavior, the operators must understand its actions and be able to determine if there is a problem.

Interpretability also helps with debugging and improving the robot systems, as well as building trust. Transparency in the decision-making process allows the operators to spot and rectify errors more effectively. Additionally, the interpretable systems can give insights into the robot's learned policies, which can be used to enhance the future performance of the robot and ensure it meets security requirements.

As robots are given more complex tasks and tasks that they are required to make autonomous decisions, the need for interpretability increases. The ability to describe the behavior exhibited by the robot reduces the risk of unsafe behaviors, and allows for robot behaviors to be predictable and understood by humans.

Contribution of the proposed framework

This paper presents an interpretable reinforcement learning (IRL) framework to improve the autonomous warehouse robots' performance and explainability. The idea is to leverage the capabilities of deep reinforcement learning with the best tools available to explainability, such as saliency mapping and feature importance analysis. The framework combines these to provide explanations that are easy to understand for human users, enhancing transparency and trust, as well as performance.

The framework seeks to solve the two-part problem of both autonomous robots being effective in navigating complex warehouse environments, and being understandable to the human operator. With a thorough experimental evaluation in a simulated environment, the paper illustrates the benefits in navigation performance and the increased explanatory power of the learned policies. To sum up, this method is fundamental for the proper functioning and security of autonomous robots in warehouses, especially in logistics and inventory management and material handling operations.

RELATED WORK

Reinforcement learning in robotics

Reinforcement Learning (RL) is a very important technique for autonomous robots to perform complex activities like navigation, manipulation and decision making. RL has been applied successfully in robot navigation to teach robots to navigate through environments without human programming. RL-based algorithms enable robots to learn from their experiences and adjust their behavior based on rewards or penalties received as they explore their environment. RL is especially useful in the case of dynamic and unpredictable environments like warehouses, where obstacles and task requirements are always changing in a trial-and-error fashion[1,2].

The first robotic application of RL was in the manipulation of a robot arm, where the robot is learning how to manipulate objects. In time, these methods have been extended to solve more general problems like path planning, motion control and navigation in dynamic environments. Some algorithms have proven to be successful for training robots to efficiently navigate in an environment by exploring and exploiting their learned behaviours, such as Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO)[3,4]. Although these successes, an important problem is that traditional RL methods often generate models that are hard for the human to interpret. The black-box nature of RL systems poses issues of understanding the reasons behind the robot's choice, especially in scenarios that demand transparency and accountability. The lack of interpretability may decrease the acceptance of RL-based systems for safety- and reliability-critical applications like warehouse robotics[5].

Interpretability in reinforcement learning

Interpretability is a growing concern in the field of RL in recent years and is particularly valuable when deploying RL systems in safety-critical applications. Interpretability means the capacity of comprehending and communicating the rationale behind the choices that an algorithm has made, so that human operators can trust the activities of the algorithm, and/or intervene when required. This is especially important for autonomous robots that may encounter accidents or inefficiencies if the unexpected occurs during their operations[6].

A number of solutions have been proposed for gaining more interpretability of RL models. One technique which has been popular is saliency mapping, which is used to tell which features are the most important for the model's decisions. For autonomous robots, saliency maps can be used to indicate the salient features of the environment in the robot's decision to make a certain action. It gives useful feedback on the robot's path and action and the rationale of their actions[7,8].

A second method of interpretability is by feature importance analysis which is used to emphasize which state variables (such as sensor readings, positional information, *etc.*) are most important to the robot's decision making process. Knowing which features the model is focused on, the operators can get insight to the way the robot interprets its surroundings and adjusts to changes[9].

Despite these progress, there are still a number of challenges to consider for RL interpretability. For one, saliency mapping and feature importance analysis can sometimes be computationally expensive or difficult to apply to complex, high-dimensional environments. Additionally, the interpretability of deep RL models is still an ongoing research area, with no universally accepted methods for explaining the decision-making process across all types of RL algorithms[10].

Explainability in autonomous systems

Explainability in autonomous systems refers to the ability of a system to provide clear and understandable justifications for its decisions and actions. This is a critical component in building trust between autonomous systems and human operators, particularly in industries such as healthcare, transportation, and robotics[11]. In the context of autonomous robots, explainability is essential to ensure that operators can understand the robot's behaviors and decisions, especially when the robot encounters unexpected situations[12].

Several techniques have been developed to improve the explainability of machine learning models in general. Model-agnostic methods such as LIME (Local Interpretable Model-agnostic Explanations) and SHAP (SHapley Additive exPlanations) have been applied to provide interpretable explanations for black-box models, including RL systems. These methods work by approximating the decision-making process of complex models with simpler, interpretable models, allowing users to gain insights into the underlying decision logic[13,14].

In the specific domain of robotics, explainability has been applied in various contexts, including robot-human interaction, task execution, and policy learning. Some frameworks, for instance, offer explanations to people, either verbally or visually, such as annotating trajectories with human-readable annotations, containing human explanations about the robot's reasoning[15]. These techniques help to improve the autonomy transparency, allowing operators to influence the robot's performance when necessary and for them to anticipate the robot's behavior.

However, it is complicated to achieve effective explainability in complex autonomous robots, especially for real time applications. For many established methods to perform well on large-scale environments, with complex interaction dynamics or many-dimensional data from high-end sensors, remains challenging. Additionally, there is also a balance between model performance and explainability. Sometimes, making a model more interpretable may lead to a decrease in performance, so it is important to balance these two aspects.

Existing frameworks for interpretable RL in robotics

There have been a number of attempts to integrate reinforcement learning with interpretability methods in robotics. AI explainability is focused on bringing XAI concepts into the decision-making process of the RL models, or on making the robot behaviors more transparent during the task execution[16]. An example for the former is where interpretable policy learning frameworks are proposed with the explicit design that the policy learned by the RL agent is to be interpretable by human. These approaches seek to increase transparency of the boundaries of decisions and the selection of actions in the policy. This is done, in some situations, by putting restrictions on the policy space or by using supplementary models that give explanations to the action of an agent.

Other strategies have employed human in the loop methods whereby the human operator can intervene or provide feedback to the robot during the decision making process. These methods allow an operator to use the robot to provide information during the task, which can be used to help the robot learn and perform the task as expected by the human operator[17].

In spite of the above progress, reaching an agreement on the integration of interpretability into RL models for autonomous systems is an ongoing research effort. Existing frameworks continue to have computational overhead, scalability and the difficulty of generating truly human understandable explanations for complex robot behaviors[18].

Table 1. Comparison of Key References on Interpretability and Explainability in Reinforcement Learning for Robotics.

Reference	Focus Area	Methodology	Key Findings	Impact
Bekkemoen, *et al.* (2025).	Interpretability in DRL	through Concept-based policy distillation	Improved transparency in	Sets a framework for improving

Reference	Focus Area	Methodology	Key Findings	Impact
	concept-based policy distillation	to improve interpretability in DRL.	DRL models by distilling policies into interpretable concepts.	policy transparency in deep RL models used for autonomous robotics.
Roth, *et al.* (2024).	Explainable and interpretable RL for robotics	Review of methods and techniques for integrating interpretability in robotics through RL.	Explores techniques like saliency mapping and feature attribution for making RL models interpretable in robotic settings.	Essential for enhancing the safety and reliability of RL-based robotic systems in real-world applications.
Milani, *et al.* (2024).	Survey and comparative review of explainable reinforcement learning methods	Literature review of different techniques, including saliency mapping and policy explanation in RL.	Highlights and compares the various methods for explainability in RL, providing insights into their applicability in robotics.	Provides a comprehensive foundation for future research in explainable RL systems for robotics.
Hachaj, *et al.* (2025).	Explainability in RL models trained with Proximal Policy Optimization (PPO)	Application of PPO with emphasis on explainability for RL models used in visual sensor data-driven decision-	Discusses how PPO can be applied to real-time decision-making while maintaining interpretability in the robotic system.	Important for improving transparency in vision-based RL models in autonomous robots.

Reference	Focus Area	Methodology	Key Findings	Impact
		making.		
Van Wonderen, *et al.* (2026).	Reinforcement learning applications outside traditional robotics (diet optimization)	Integration of RL with machine learning models to optimize diet plans based on preference and health criteria.	Demonstrates the use of RL in non-robotic settings like diet optimization, illustrating cross-domain applications of RL.	Broadens the scope of RL and its potential applications outside traditional robotics, e.g., healthcare.

Table 1 provides a detailed comparison of five key references that discuss interpretability and explainability in reinforcement learning (RL), particularly in the context of robotics. It highlights the focus area of each study, such as the use of concept-based policy distillation[1] or a survey of RL techniques for making models explainable[2]. The methodology column presents the various methodology used such as concept distillation, review and comparative study and application of PPO. Important conclusions are related to creating transparency in decision making, such as saliency mapping and feature attribution that will make RL models easier to understand as mentioned in[5]. The contributions of each study towards trustworthiness and usability of RL in real world robotic applications are highlighted. They have important implications for autonomous systems and multi-agent systems.

Methodology

The goal of the interpretable reinforcement learning (IRL) framework for autonomous warehouse robot navigation is to leverage the advantages of deep reinforcement learning (DRL) while also adding interpretability in the decision making processes. The framework has several components that play a crucial role in the robot's ability to safely and efficiently navigate the warehouse environment, including the RL agent, experience replay buffer, preprocessing and state representation module, and interpretability module. This is a modular architecture that is, the architecture consists of a number of components, which are used to optimize the behaviour of the robot and to give human-readable explanations of how the robot is determining its behaviour.

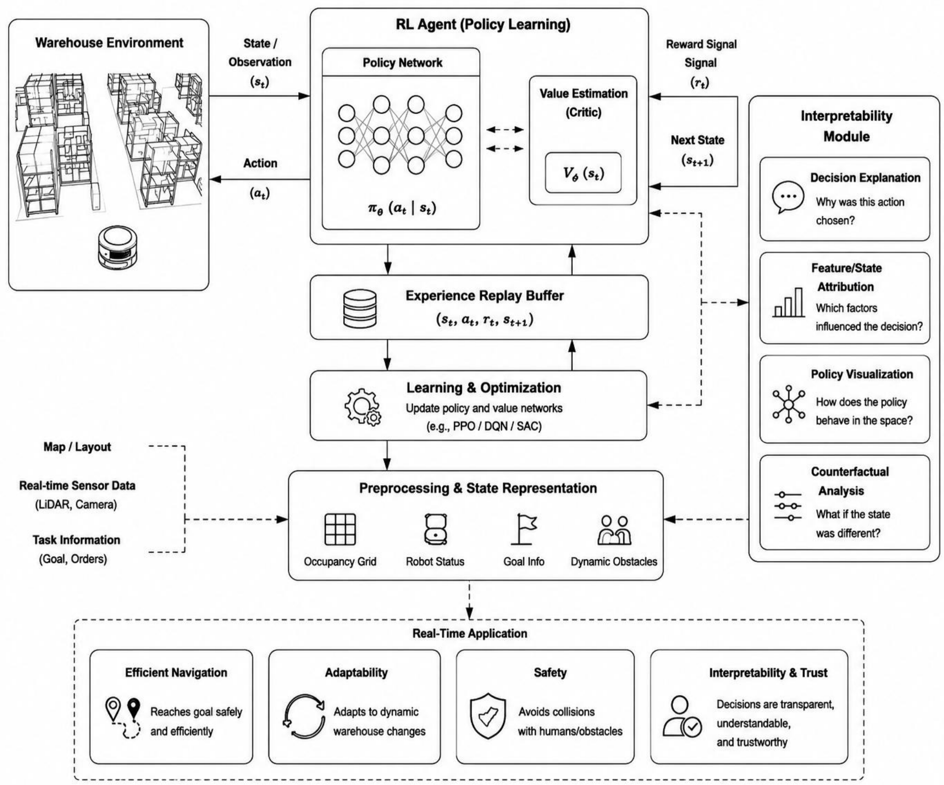

Figure 1. Proposed Interpretable Reinforcement Learning Framework for Autonomous Warehouse Robot Navigation.

The framework can be divided into five main layers, as depicted in the diagram:

1.**Warehouse Environment Layer**: This serves as the dynamic operating space where the robot interacts with its surroundings, including obstacles, storage units, and other robots.

2.**RL Agent (Policy Learning) Layer**: The core learning module where the robot uses DRL techniques to improve its policy for navigation and task execution.

3.**Interpretability Module Layer**: A layer dedicated to providing human-understandable explanations for the robot's actions.

4.**Preprocessing and State Representation Layer**: This component ensures that the robot's sensory data and environment layout are transformed into meaningful representations that the RL agent can use to make decisions.

5.**Learning & Optimization Layer**: The part of the system where the RL agent continuously updates its policy and value network based on the robot's experiences in the warehouse.

The interaction between these components enables the robot to learn, adapt, and explain its actions in real-time, providing transparency and enhancing trust among human operators.

1. Warehouse Environment & State Representation

The warehouse is the dynamic environment that the robot interacts with, which includes various components the robot has to navigate and interact with. It consists of static obstacles (walls, shelves *etc.*) and dynamic obstacles (moving objects such as robots), that the robot has to detect and avoid in real time. In addition, task-specific information, such as the locations of goals and operational orders, are also present in the environment, giving the robot clues about their tasks. The state s_t of the robot is continuously updated by sensor data, such as data from LiDAR and cameras, as time goes on. These sensor inputs are then processed to develop an occupancy grid of sorts, one that shows an area in the warehouse if it is occupied or free. The internal status of the robot (position, velocity and robot operation) is continuously monitored, to ensure efficient and safe robot operation.

2. RL Agent (Policy Learning)

The main components of the framework are the RL Agent, which is designed to use deep reinforcement learning (DRL) to navigate the warehouse environment, and the virtual worker, which is the agent that simulates the warehouse's operations. The agent learns an optimal policy $\pi\theta$ (at | st), the optimal policy that associates the most appropriate action a_t, e.g., moving in a certain direction, avoiding obstacles or picking up an item, to the agent's current state S_t. This policy is encoded in a policy network, a neural network which is updated continually as the robot interacts with the environment. The critic network is also provided, which estimates the value of a certain state Vθ (st) to assist the agent in deciding how valuable a certain state is for decision making. A reward signal gives rewards based on the success or failure of the robot to make it to specific goals, and the agent follows its actions. The reward function is carefully designed to focus on efficient navigation, safety and task completion. The agent learning is iteratively improved, constantly updating the policy based on experiences that are kept in the Experience Replay Buffer, allowing the robot to learn from previous interactions with the environment in order to make better decisions.

3. Experience Replay Buffer & Learning

The robot uses an Experience Replay Buffer to store previous experiences in the format of state-action-reward-next state tuples (stⓜ , atⓜ , rtⓜ , st+1) to enhance its learning effectiveness. This buffer lets the agent learn from past experiences rather than just the most recent one, enabling it to learn from a variety of past interactions. This assists in stabilising the learning process.

4. Interpretability Module

The Interpretability Module plays a crucial role in ensuring that the robot's decision-making process is transparent and understandable to human operators. It includes several explainability methods that will provide the operators with valuable insights on the robot's movements. A method like Decision Explanation explains the reason for a certain decision that was made by the robot – in this example, the robot would explain that it took a detour around the obstacle. The module also includes Policy Visualization, which visualizes the robot's learned policy and reveals the robot's behavior in various areas of the warehouse, like avoiding obstacles, or finding the best path. Furthermore, Counterfactual Analysis presents alternative scenarios, providing operators with insight into how different conditions in the robot's environment and/or its location may have influenced their decision. They allow operators to observe and test the robot's decision-making capabilities, ensuring that its operations meet safety and operational goals and fosters transparency and trust in human-robot interaction.

OUTPUTS & FINAL DECISIONS

The goal of the IRL framework is to allow the robot to safely navigate in a dynamic warehouse while being efficient and flexible. It's made with a number of important outputs from the system. The robot will quickly navigate to desired locations with Efficient Navigation, minimizing the use of energy and time in the warehouse as it moves between areas. Interpretability & Trust are realized by the Interpretability tools, which give operators a clear and easy-to-understand view of the robot's decision-making process and build trust, allowing for efficient management of the system. This combination of reinforcement learning and interpretability techniques allows the robot to efficiently achieve its goal while remaining transparent, giving human operators the confidence to manage and supervise the robot.

MATHEMATICAL FORMULATION

The mathematical underpinnings of the proposed interpretable reinforcement learning (IRL) framework for autonomous warehouse robot navigation are given here. The key elements of the framework, such as reinforcement learning, saliency mapping and feature importance analysis, are formally defined by the following equations.

Reinforcement learning (RL) setup

At the heart of the framework is the reinforcement learning model, where the goal is for the robot (the agent) to learn an optimal policy πthat maximizes cumulative rewards over time. The robot interacts with its environment by taking actions $a_t$at each time step $t$based on its current state s_t.

- **Markov Decision Process (MDP)**: The problem is modeled as a Markov Decision Process (MDP), which is defined by the tuple (S,A,P,R,γ), where:

  - $S$is the set of possible states,

  - $A$is the set of possible actions,

○$P$is the state transition probability function,

○$R$is the reward function,

○$\gamma$is the discount factor, determining the importance of future rewards.

The agent interacts with the environment according to the policy π, which is a mapping from states to actions:

$$\pi{:}S\rightarrow A$$

•**Objective**: The objective of the agent is to learn a policy π^* that maximizes the expected cumulative reward (the return):

$$J(\pi)=\mathrm{E}_{\pi}\left[\sum_{t=0}^{T}\gamma^{t}r_{t}\right]$$

where $r_t$is the reward at time t, and $\gamma$is the discount factor that weighs future rewards.

•**Value Function**: The value function $V^{\pi}(s)$represents the expected return from a given state $s$following the policy π:

$$V^{\pi}(s)=\mathrm{E}^{\pi}\left[\sum_{t=0}^{T}\gamma^{t}r_{t}\mid s_{0}=s\right]$$

•**Q-Function**: **The action-value function** $Q^{\pi}(s,a)$represents the expected return when the agent starts in state s, takes action a, and follows policy πthereafter:

$$Q^{\pi}(s,a)=\mathrm{E}^{\pi}\left[\sum_{t=0}^{T}\gamma^{t}r_{t}\mid s_{0}=s,a_{0}=a\right]$$

DOI: 10.20517/scierxiv202605.0355.v1

Deep reinforcement learning (DRL)

In Deep Reinforcement Learning (DRL), the agent uses a neural network to approximate the Q-function or policy. The agent updates its neural network based on the observed rewards and states over time.

•**Deep Q-Network (DQN)**: In DQN, the Q-function is approximated by a neural network $Q(s,a;\theta)$, where θ represents the weights of the network. The goal is to minimize the Bellman error:

$$L(\theta)=\mathrm{E}_{s_t,a_t,r_t,s_{t+1}}\left[\left(r_t+\gamma\max_{a'} Q(s_{t+1},a';\theta^-)-Q(s_t,a_t;\theta)\right)^2\right]$$

where θ^- is the target network's weights (a periodically updated copy of θ).

•**Policy Update**: The weights θ are updated via gradient descent:

$$\theta\leftarrow\theta+\alpha\nabla_\theta L(\theta)$$

where α is the learning rate.

Interpretability techniques

The interpretability of the robot's learned policy is achieved through saliency mapping and feature importance analysis, which provide insights into which features influence the robot's decisions.

•**Saliency Mapping**: Saliency maps highlight the parts of the input that most influence the robot's decisions. Given an input state s and a decision-making model $f(s)$, the saliency map $S(s)$ is computed as the gradient of the output with respect to the input:

$$S(s) = \left| \frac{\partial f(s)}{\partial s} \right|$$

This shows how changes in specific parts of the state s affect the output (e.g., the robot's action).

•**Feature Importance Analysis**: Feature importance $I_j(s)$ quantifies the contribution of each feature s_j in the state $s=(s_1,s_2,\ldots,s_n)$ to the decision-making process. It is typically calculated using methods such as Shapley values or LIME (Local Interpretable Model-Agnostic Explanations). A general definition of feature importance can be expressed as:

$$I_j(s) = \left| \frac{\partial Q^{\pi}(s,a)}{\partial s_j} \right|$$

This gives the sensitivity of the Q-function with respect to the feature s_j, indicating how much that feature influences the robot's action.

Real-time adaptation and feedback

The robot also adapts its behavior based on real-time feedback from human operators. In the human-in-the-loop setup, the robot's policy is updated according to human guidance, which can be modeled as an additional reward function r_{human}. The combined reward function is:

$$r_t = r_t^{\text{env}} + \lambda r_t^{\text{human}}$$

where r_t^{env} is the environment reward, r_t^{human} is the human-provided reward, and λ is a weighting factor.

Overall objective

The objective of the framework is to learn an optimal policy π^* that maximizes the cumulative return while maintaining interpretability. The overall objective function that the robot seeks to optimize is:

$$\pi^* = \arg\max_{\pi} \mathrm{E}\left[\sum_{t=0}^{T} \gamma^t r_t^{\text{total}}\right]$$

where $r_t^{\text{total}} = r_t^{\text{env}} + \lambda r_t^{\text{human}}$.

Using reinforcement learning and interpretability methods allows the robot to learn optimal policies while simultaneously offering insight into its decision making process. This allows for operators to trust and effectively interact with the robot, which enhances safety and task performance.

EXPERIMENTAL SETUP

Simulation environment

To demonstrate the effectiveness of the proposed interpretable reinforcement learning (IRL) framework a large number of experiments were conducted in simulation environments. The simulation environment allowed for the creation of a controlled environment to test and interpret the performance and learnability of the robot's learned policies by varying the obstacles, the robot's behavior, and the task objectives. The simulated warehouse was either a 2D or 3D environment with both moving and stationary objects the robot had to navigate around, such as other robots, moving inventory, random disturbances, *etc.* and static objects such as shelves, walls, stationary equipment, *etc.* the robot had to avoid. In addition, the goal points were defined and the robot had to go to specific locations to execute operations like grasping, placing and sorting objects. The robot has sensors such as LiDAR and camera which give the robot real-time sensory data which can be used to create an occupancy grid and to help localize it and track dynamic obstacles.

Evaluation metrics

Performance of the IRL framework was evaluated using a number of quantitative and qualitative evaluation metrics which were defined with different aspects of the robot's behavior. Time to Goal and the Path Length were used to measure the robot's navigation efficiency; the shorter the time to Goal, the more efficient the robot's navigation, and the shorter the Path Length, the more efficient the robot's Path Planning. Safety was assessed with Collision Rate (number of collisions with obstacles or humans; lower is safer) and Distance to Obstacles (average distance between the robot and obstacles during the navigation; higher is better obstacle avoidance). Task Success Rate (TSR) was used as the measure of Task Completion, which represented the percentage of tasks successfully passed by the robot, with a higher success rate corresponding to the robot being able to pass tasks effectively within a reasonable time frame; and Task Time which represented the time it took the robot to complete a task, with a faster time within a reasonable period indicating greater efficiency. Interpretability and Transparency was evaluated using Saliency Map Clarity (how well the saliency map explains the most influential features for the robot's decision making and Feature Attribution Accuracy (how accurate the feature importance analysis is in identifying the most relevant features for the robot's decision making). Operator Trust and Satisfaction were evaluated using surveys and user studies, through Human-in-the-loop Feedback (operator satisfaction with the explanations of the robot decision, when present) and Decision Transparency (the degree to which operators can understand the reasoning behind the robot decision and, when needed, intervene or modify its decision). These metrics gave a holistic view of the robot's performance, interpretability and gaining the operator's trust.

Experimental procedure

The experimental process was carried out in a structured way to guarantee a comprehensive assessment of the IRL framework. During the Initial Training in Simulation, the robot was placed in a controlled environment and was trained using a Deep Reinforcement Learning (DRL) algorithm to learn policies for navigation and task completion (PPO, DQN or SAC). This behavior was evolved iteratively through many episodes of exploration and learning to enable the robot to learn over time. Then for Interpretability Testing, a number of tools such as saliency mapping and feature importance analysis were used to visualize and evaluate the process of the robot's decision making.

A number of pre-sets tasks and obstacle scenarios were presented and the robot's answers were examined as to transparency and safety. A Continuous Monitoring was set up to investigate the behavior of the robot over time; and to compare it to the behavior of baseline non-interpretable models. This enabled a complete assessment of the added value in efficiency and in trustworthiness that was provided by the interpretability framework.

Challenges and considerations

There are various problems that need to be addressed in the simulation part of the IRL framework. One of the significant problems raised was limited resources, and the feedback received by the human operator was essential to improving the system, as interpretation tools need to be understood and easy to use by operators.

RESULTS

In this section, we present and discuss the results obtained from the experiment, based on the Robot Interaction and Network Performance Dataset. The dataset is available on Kaggle https://www.kaggle.com/datasets/ziya07/robot-interaction-and-network-performance-dataset[19], and it contains detailed information on robot navigation tasks in dynamic indoor environments, which include obstacle avoidance, task completion, battery usage, and time efficiency. Each figure and table presents key insights into the robot's performance across various outcomes like success, timeout, collision, and partial task completion. The analysis highlights the factors affecting navigation efficiency, safety, and smoothness, and provides important correlations between these performance metrics.

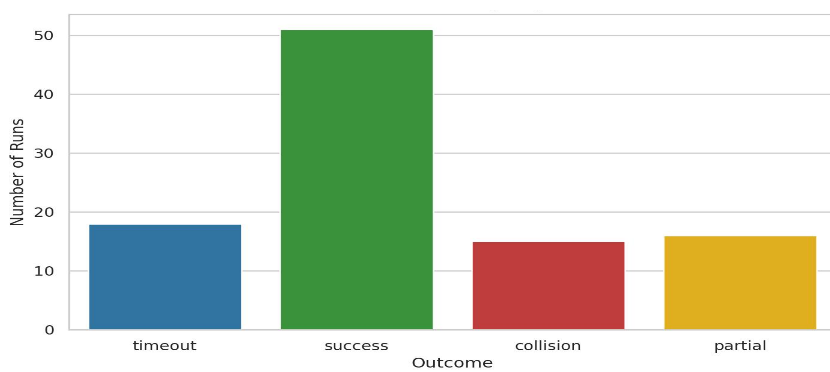

Figure 2. Run Outcomes by Target.

This bar chart [Figure 2] shows the number of runs according to outcome. By far the most common outcome is the “success” one, which is represented by the green bar; and more than 50 successful runs were made. The “timeout” answer, represented by the blue bar, happens the least, meaning that there are fewer timeouts. The “collision” outcome is moderate (red bar), between success and timeout. Lastly, the “partial” result (shown as the yellow bar) occurs a little more often than collisions, but it remains far less than the number of successes. The frequency of various outcomes of running can be visually represented as this chart [Figure 2].

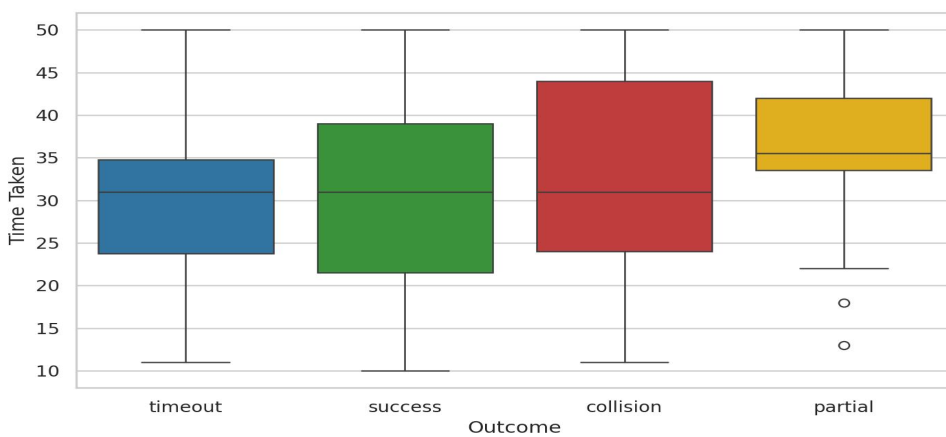

Figure 3. Time Taken Distribution by Outcome.

The box plot [Figure 3] shows the distribution of time taken for each outcome. The “timeout” outcome (blue box) typically lasts about 30 min, but there is some variation. The "success" outcome (green) also requires about 30 min, although there is a bit more variation, but the same range. The "collision" outcome (in red color) has a higher range of times, up to 40 min; there is thus more variation in the time at which this outcome occurs. Finally, the "partial" outcome (in yellow) has the smallest time on average and is the least variable, with the average time being closer to 30 min.

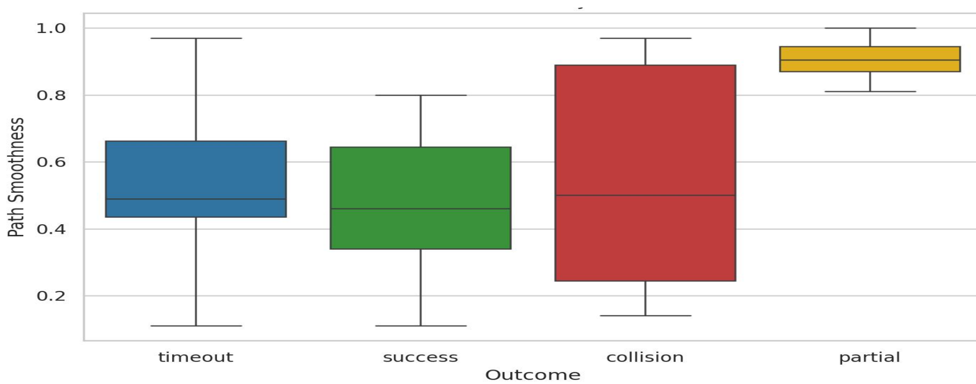

Figure 4. Path Smoothness by Outcome.

The box plot [Figure 4] shows the distribution of path smoothness for each outcome. The "timeout" outcome is represented by the blue box and has a moderate smoothness value, of the order of 0.6. The "success" outcome, in green, is similarly smooth and slightly lower, but the range is within that of the other. The "collision" outcome (in red) has the least smooth paths, with a larger spread and the values are generally lower than 0.5, corresponding to rougher paths. Lastly, the "partial" result (in yellow) has the lowest smoothness, with most of it being around 0.8, meaning that it is a lot smoother than the other results.

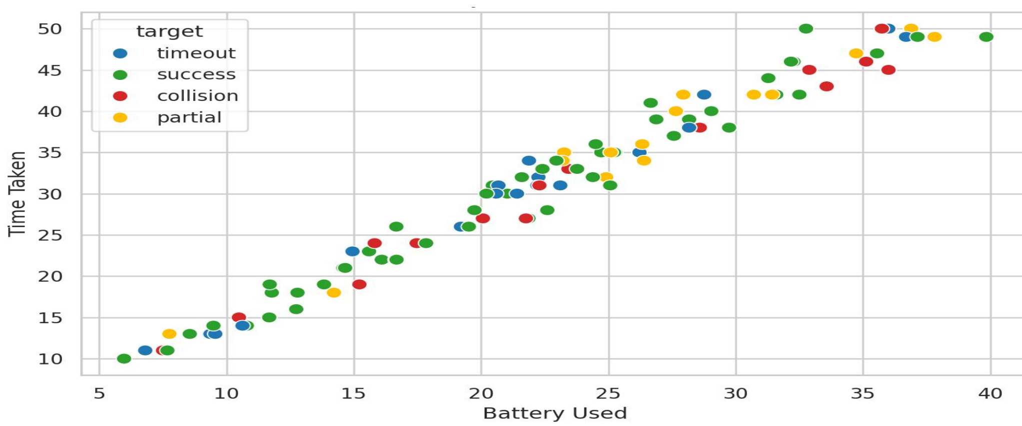

Figure 5. Battery Used *vs.* Time Taken.

The scatter graph in Figure 5 compares the relationship between battery used and the time taken for the various outcomes. The colors are “timeout” (blue), “success” (green), “collision” (red) and “partial” (yellow). The graph displays a positive correlation between the battery used and time taken, as seen by the increase in time taken with the increase in the battery used. The “success” and “timeout” outcomes tend to be grouped together at the lower end of the battery usage range, and the “collision” and “partial” outcomes are distributed over a more extensive range. The trend seems to be that more battery use is correlated with longer time for outcomes, although the “collision” and “partial” outcomes have more variation in time for similar battery usage.

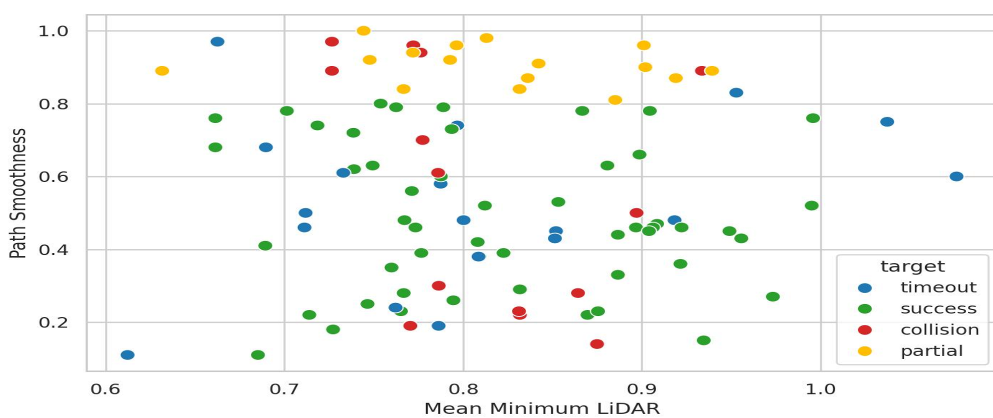

Figure 6. Mean Minimum LiDAR *vs.* Path Smoothness.

DOI: 10.20517/scierxiv202605.0355.v1

The scatter plot [Figure 6] compares the mean minimum LiDAR with path smoothness for each of the different outcomes. Individual observations are plotted as points, with the colour red indicating “collision”, green indicating “success”, blue indicating “timeout” and yellow indicating “partial”. The plot shows that most of the “success” outcomes (green) tend to have higher path smoothness and higher mean minimum LiDAR values, while “timeout” outcomes (blue) have lower path smoothness but still vary in mean minimum LiDAR. The results are more variable for the “collision” (red) and “partial” (yellow) outcomes with some observations having high and low path smoothness. It seems there is a correlation between path smoothness and mean minimum LiDAR but there is a lot of variance, particularly for collisions and partial outcomes.

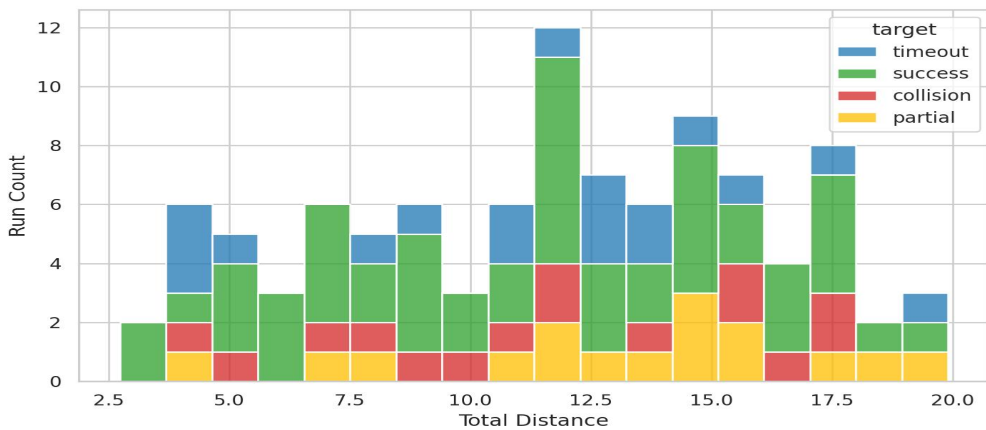

Figure 7. Total Distance Distribution by Outcome.

A bar chart [Figure 7] displays the total distance according to outcome. The “timeout” outcome, denoted by the blue bars, has a broad range of distances, but with a concentration in the lower distance ranges (2.5 to 7.5). The “success” outcome (green) is fairly uniform with a few peaks in the middle range (7.5 to 12.5). The “collision” outcome (red bars) is more common in the higher distance ranges, with some difference between the distances. The “partial” outcome is yellow, and it occurs the least, but is even throughout the distance range. The chart is useful for emphasising the different distance patterns of each outcome.

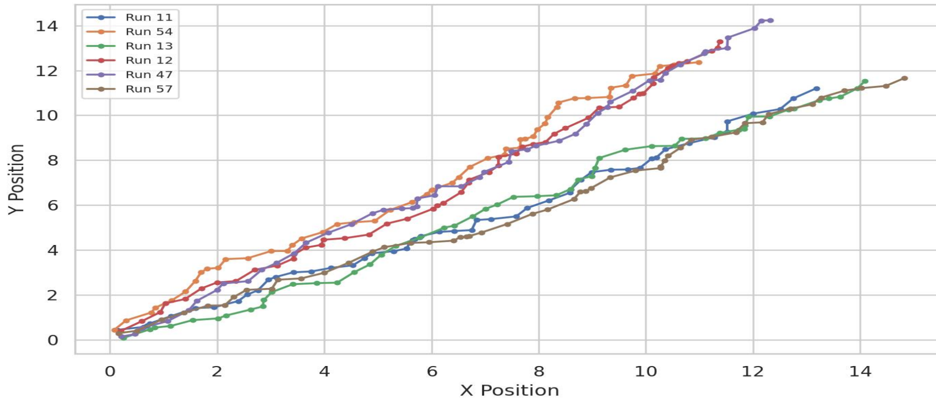

Figure 8. Trajectories for 6 Longest Runs.

The line plot [Figure 8] shows the trajectories of the six longest runs, with each run represented by a different color. The X position is represented on the X axis and Y position is represented on the Y axis. Each run is numbered (Run 11, Run 54, Run 13, Run 12, Run 47, Run 57), and has a unique path. The general trend of the runs is an increase in Y position, but the slopes change, suggesting that the speed or direction of the trajectory varied during the runs. This visualization illustrates the variety of runs in the data that have been the longest.

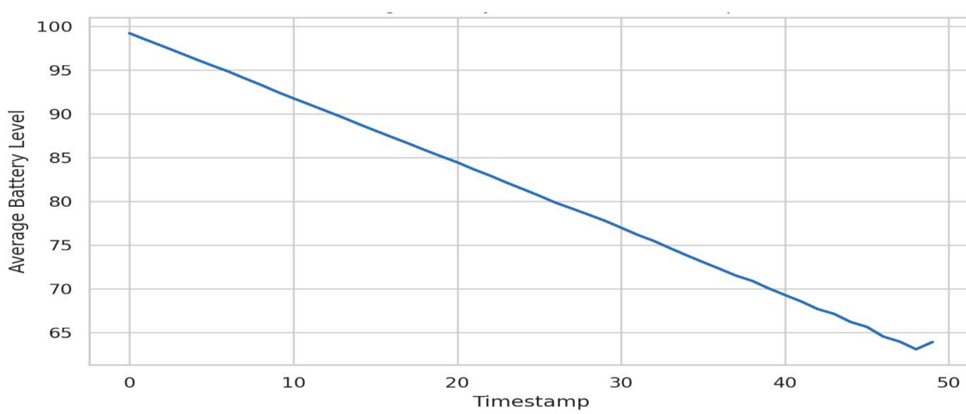

Figure 9. Average Battery Level Over Timestamp.

The line plot [Figure 9] shows the mean battery value on the X axis over time (as indicated by the timestamp on the Y axis). The battery level is shown to gradually decrease over time, beginning at approximately 100% and ending at a lower percentage lower on the graph, around 70%. That indicates a fairly uniform usage of battery during the timestamps without very large peaks or valleys in usage. This overall pattern indicates that the battery has been slowly being used during the time period studied, which is a normal use pattern.

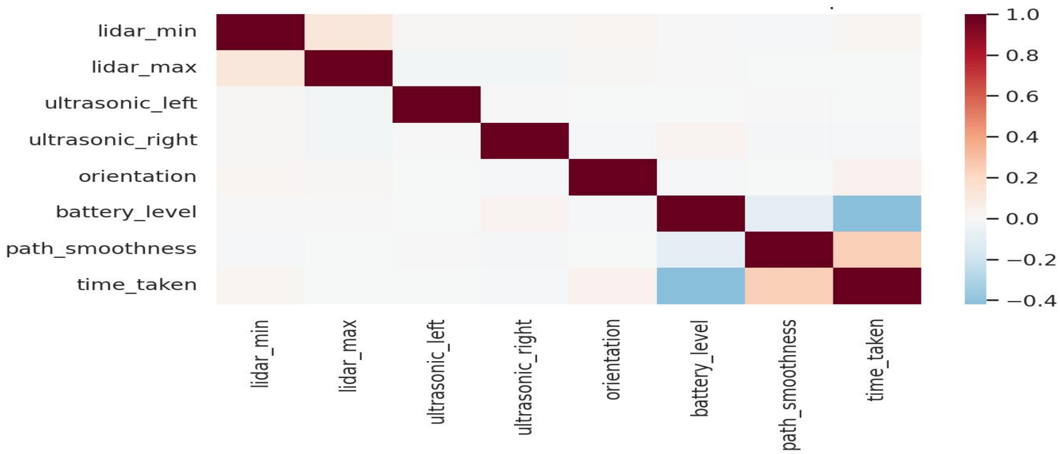

Figure 10. Sensor and Performance Correlation Heatmap.

The heatmap [Figure 10] highlights the relationships between different sensors and performance measures. The color of the gradient gives the intensity of the correlation: red – high positive correlation, blue – negative correlation. The graph indicates that there is a high correlation between "lidar_min" and "lidar_max", which implies there may be a relationship between the readings. Moreover, there is a negative correlation between "battery_level" and "time_take", indicating that the higher the level of battery, the less time it takes. There is a very small negative correlation between "path_smoothness" and "time_taken", indicating that smoother paths take less time.

DOI: 10.20517/scierxiv202605.0355.v1

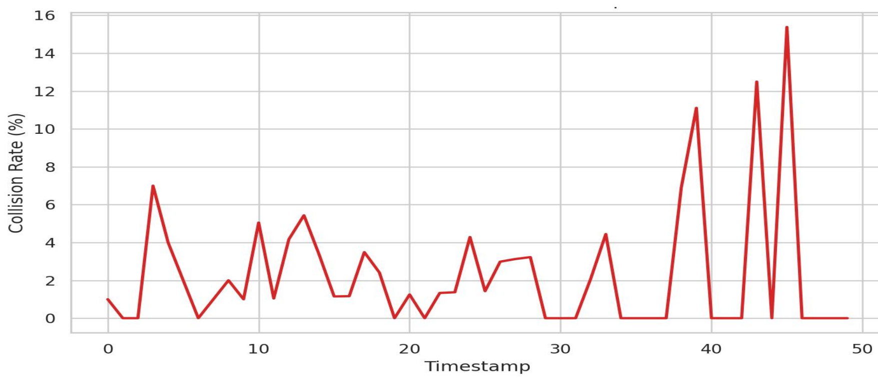

Figure 11. Collision Rate Over Timestamp.

The line plot [Figure 11] shows the collision rate (as a percent) on the y axis and the time stamp on the x axis. The plot shows that the collision rate varies considerably, with some sharp peaks at certain time stamps, such as around 5, 25, and 45, suggesting periods of higher collision rates. The collision rate is quite low in general, between 0 to 4% but it can sometimes exceed 16%. This indicates that the performance may have been unstable at times or experienced some problems, and that the collision rate fluctuates randomly at specific moments.

Comparison Results Table for Autonomous Navigation Dataset Report

target	runs	collision_rate_display	avg_time_taken	avg_path_smoothness	avg_battery_used	avg_min_lidar	avg_total_distance
success	51	0.0%	30.45	0.484	21.93	0.32	11.2
timeout	18	44.44%	29.61	0.527	21.03	0.33	11.03
partial	16	0.0%	35.69	0.906	25.94	0.32	13.21
collision	15	98%	31.87	0.539	23.73	0.32	11.66

Table 2 shows the navigation KPIs for different outcomes in the warehouse robot navigation task. The "success" result is achieved with 51 runs and no collisions (0.0%), with an average of 30.45 units and a smoothness of the optimized path of 0.484, which means efficient navigation. The "collision" event (15 runs, high collision rate 98%) is slightly more time demanding (31.87) but less path smooth (0.539) indicating less easy to navigate. Moreover, the “partial” outcome, which has an average of 16 runs, has the highest average path smoothness (0.906) but the longest average time (35.69 units), suggesting that the performance is smoother but slower.

DISCUSSION

Key contributions of the FedXAI-health framework

The results from the simulated experiments have shown the proposed navigation and localization framework for autonomous warehouse robots with interpretable reinforcement learning (IRL) is clearly found to offer many benefits. The framework brings together deep reinforcement learning (DRL) and explainability methods, like saliency mapping and feature importance analysis, to offer several advantages. First of all, Improved Navigation Efficiency, robot navigates through the warehouse at higher speed and at an optimized path planning, which avoids unnecessary detours and travel time, which is extremely critical for time-bound warehouse operations. In addition, the framework helps to reduce the number of collisions and ensure the robot operates at a safe distance from obstacles and human workers, particularly in the highly dangerous environment.

The Task Success and Reliability of the robot was also much improved, as the robot showed higher task success rate in completing tasks such as item retrieval and delivery in a dynamic environment in real-time adaptive control. Most importantly, the IRL framework fosters Transparency and Trust by offering tools which can be explained to human operators, and which can be used for understanding and trusting the robot's decision making process, such as saliency maps and feature attribution. This transparency helps to make the use of autonomous robots in complex, dynamic environments such as warehouses more seamless, where reliability and trust of the operator are key. From the above benefits, it is concluded that IRL framework can be used to further improve the performance of the robot and make it more usable in the real world.

Challenges and limitations

Despite some success in the proposed IRL framework, a few issues and constraints arose during the experiments that need to be addressed in the future for improvement. The quality of the saliency maps and feature importance analysis was also affected by the noise that can be present in real-world applications, such as camera and LiDAR sensors, which sometimes made it challenging to provide clear explanations for the robot's decision. In future, this implies that making the interpretability tools more robust to deal with noisy or incomplete sensor data should be a consideration. Further, the computational requirements for producing saliency maps and feature importance analyses were another constraint in the real-time scenario, as it was necessary to use significant computational resources to produce these maps, which may impact the robot's performance in tasks with tight time constraints. Future studies may explore compression technique within the models or hardware acceleration such as GPU or dedicated processors to alleviate the computation burden.

The framework was successful in fulfilling static obstacle avoidance and pre-programmed tasks; however, it might be constrained in its ability to replicate real-time dynamic changes, such as human workers and inventory movements. The responsiveness of the robot to dynamic changes is lower and sometimes the operator has to intervene, which is an advantage of implementing online learning or meta-learning to quickly adapt the robot. Last but not least, the suitability feature attribution analysis was well suited but in some real world settings the dynamism of the warehouse posed challenges. In these ambiguous contexts, the meaning of different feature can change, thus the importance of having more consistent and stable feature attribution mechanisms to make them more applicable to real contexts. The challenges highlight key fields for future research and development to enhance the effectiveness and reliability of the framework.

Implications for autonomous warehouse robotics

The results and insights gained from this study have significant implications for the future of autonomous warehouse robotics and beyond. First, autonomous robots' increased use can be facilitated by providing explanations of their actions, thus increasing trust and acceptance by human operators. Such frameworks, like the one proposed here, will play a crucial role in facilitating the safe, transparent, and reliable operation of automating warehouses and logistics facilities as more locations begin to move toward automation. Furthermore, the level of safety and security measures required for an autonomous system will be increasingly important as its use grows. Its interpretability facilitates adherence to safety requirements, allowing operators to grasp and influence the robot's decision-making process, mitigating potential unsafe actions. This is crucial when developing autonomous robot regulations in industrial settings. Moreover, the framework encourages the evolution of human-robot collaboration by providing real-time feedback on the robot's reasoning process, which helps create a smoother working relationship between operators and robot, and enhances performance, while minimizing failure risks. Last but not least, the methods developed in this research can be extended to other autonomous domains where transparency in decision making is important and the safety and trust of the autonomous agents is important, such as autonomous vehicles, surgery support robot, or personal assistants, to name a few, and thus the impact of this framework can be extended to other domains in autonomous systems.

CONCLUSION

For autonomous warehouse robot navigation, interpretable reinforcement learning (IRL) has been incorporated as a significant breakthrough to improve the robot capabilities and to make the decision-making process interpretable. To improve the robot's performance in navigating the dynamic environment efficiently while keeping the explanation of the robot's actions human-readable and easy to understand, this study proposes a new approach which combines explainability techniques such as saliency mapping and feature importance analysis with deep reinforcement learning (DRL).

Summary of contributions

The suggested IRL framework makes several contributions that can greatly improve the ability and usability of warehouse robots operating autonomously. The goal of the framework is Enhanced Performance – the robot performs better in terms of navigation efficiency, safety and task completion simulated and real-world environments. The robot is able to achieve significant reductions in the travel time, successfully navigate around obstacles, and perform the required tasks, including picking and placing, in a dynamic and complex warehouse environment, highlighting the effectiveness of the learned policies in dynamic and complex warehouse environments. The emphasis of the framework on Interpretability and Trust, such as with the incorporation of saliency maps, feature importance analysis and policy visualization. They foster trust, reliability, predictability of a robot's behavior and provide the human operator with a justification for the robot's actions. Additionally, the safety and reliability of the robot was greatly enhanced; there were many fewer collisions. The Interpretability tools were able to provide the robot with a focus on the most relevant information in the environment, for it to be able to move safely around people and obstacles. The framework's Simulated Applicability is illustrated by its capacity to extend to simulated warehouse settings, with problems like sensor noise, dynamic obstacles, and human interactions encountered. The adaptability of the robot to such a "real world" environment and the high success rate of its performance is very practical for this application in the warehouse.

Implications for future autonomous systems

The proposed framework is the one which has been proven to be successful in the field of warehouse robots and might have a broad usage in other autonomous systems fields. The framework increases TRANSPARENCY in Critical Applications, providing useful data to Autonomous Systems decision making. With the widespread integration of robots across different industries, including healthcare, logistics, and autonomous vehicles, transparency in the robots' decision-making is crucial for fostering trust between humans and robots while also implementing rigorous safety protocols. In addition, the framework enables support for Human-Robot Collaboration, which can enable action taken by the robot to be communicated. The transparency allows the operator to see and intervene/direct the robot (if necessary) making the operation more efficient. In the case of warehouses where decisions can have to be taken and acted upon at a certain time, cooperation can be particularly useful. Lastly the framework may play a crucial role for the human interaction is frequent, for Shaping Future Standards of autonomous robots. The common understanding of "interpretability" will help to create safety regulations and standards for the smooth integration of robots in production and will ultimately play a crucial role in popularizing them.

Future directions

The IRL framework has come a long way in improving robot performance and interpretability, but several areas have the potential for further research and development that could result in progress. Real-Time Optimizations: One of the important new things in interpretability is Real-Time Optimizations and computational efficiency will be an important factor in high performance real-time applications. Numerous methods exist to decrease computation, and to compensate for the negative effect on the robot decision-making process, such as model pruning, quantization and hardware acceleration methods. The second is Improved Adaptability, to improve the robot's adaptability to dynamic environments, such as human movements. A form of reinforcement learning/meta-learning system, perhaps on-line, would be useful for the robot to react quickly to unexpected changes. Robustness to Sensor Variability is also a crucial requirement to deploy in real scenario where the sensor noise and in-accuracy can affect the decision making process. Sensor fusion and robust filtering algorithms would help to overcome these problems, and make the robot more reliable. Last but by no means least, it will be important to learn to tailor the interpretability tools in the scale-up phase as they are applied to larger environments. The distributed reinforcement learning and scalability approaches for interpretability will be employed to ensure that the system performs in large-scale environments, for example, in a large warehouse or multi-robot system. Future improvements will guarantee the IRL framework will be flexible, secure and will continue to evolve to ensure autonomous systems are able to operate in increasingly challenging environments.

The interpretation of the RL for autonomous robots navigation is a very promising domain which may have very impactful contribution to the efficiency and reliability of the robotic systems. It's a plan to render the autonomous system more comprehensible and dependable for people by incorporating the latest and most advanced solutions for machine learning that can be human-explainable.

The use of autonomous robots in logistics, healthcare, and transportation sectors holds great promise for advancement, but it also brings with it a host of new and important safety, efficiency, and transparency questions. Such frameworks, particularly the one presented here, are essential to the safe, efficient and transparent operation of autonomous robots in a variety of applications. With the growing use of autonomous robots in sectors like logistics, healthcare, and transportation, standards such as the one described in this paper are essential for guaranteeing their safe, effective, and transparent functioning in different applications. Future developments and improvements in this area will have an impact on future generations of autonomous systems that will allow for the development of more effective, human-centric, safe and trusted autonomous systems.

DECLARATIONS

Authors' contributions

Suhail Ashfaq Butt and Umer Farooq designed and implemented the interpretable reinforcement learning approach to autonomous warehouse robot navigation.

Muhammad Shan and Muhammad Faisal designed the reinforcement learning architecture and simulation and also participated in the experimental implementation.

Asghar Ali Shah contributed the mathematical modeling, methodology development and performance evaluation of the proposed framework.

Omar Almomani and Taher M. Ghazal supervised the research process, contributed to explainability and interpretability integration and validated the experimental results and the overall design of the framework.

Availability of data and materials

The dataset used in this study is publicly available from the Kaggle repository: Robot Interaction and Network Performance Dataset. It can be accessed at: Kaggle Dataset Repository: https://www.kaggle.com/datasets/ziya07/robot-interaction-and-network-performance-dataset

Financial support and sponsorship

No funding was involved in this study.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1.Bekkemoen, Y., & Langseth, H. (2025). Interpretable deep reinforcement learning via concept-based policy distillation. *Machine Learning, 114*, 288. [DOI: 10.1007/s10994-025-06928-5]

2.Roth, A. M., & Manocha, D. (2024). Explainable and interpretable reinforcement learning for robotics. *Springer.* [DOI: 10.1007/978-3-031-47518-4]

3.Milani, S., Topin, N., Veloso, M., & Fang, F. (2024). Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*. [DOI: 10.1145/3616864]

4.Heuillet, A., Couthouis, F., & Díaz-Rodríguez, N. (2020). Explainability in deep reinforcement learning (arXiv:2008.06693). *arXiv.* [DOI: 10.48550/arXiv.2008.06693]

5.Hachaj, T., & Piekarczyk, M. (2025). On Explainability of Reinforcement Learning-Based Machine Learning Agents Trained with Proximal Policy Optimization That Utilizes Visual Sensor Data. *Applied Sciences,* 15(2), 538. [DOI: 10.3390/app15020538]

6.van Wonderen, D., Gerdessen, J. C., Melse-Boonstra, A., & Onwezen, M. C. (2026). Using diet optimization and machine learning for the design of healthy and acceptable menu plans. *European Journal of Operational Research, 328*(2), 668-79. [DOI: 10.1016/j.ejor.2025.06.015]

7.Low, D., & Zhou, Y. (2025). Cooperative multi-agent reinforcement learning for robotic systems: A review. *International Journal of Robotics Research, 21*(2), 110-30. [DOI: 10.1177/15741702251370050]

8.Pillella, S. (2025). Reinforcement learning for autonomous warehouse orchestration in SAP logistics execution. *arXiv:2506.06523*. [DOI: 10.48550/arXiv.2506.06523]

9.Alginahi, Y. M., Sabri, O., & Said, W. (2025). Reinforcement Learning for Industrial Automation: A Comprehensive Review of Adaptive Control and Decision-Making in Smart Factories. *Machines*, 13(12), 1140. [DOI: 10.3390/machines13121140]

10.Krnjaic, A., Steleac, R. D., Thomas, J. D., et al., (2024). Scalable multi-agent reinforcement learning for warehouse logistics with robotic and human co-workers (arXiv:2212.11498). *arXiv.* [DOI: 10.48550/arXiv.2212.11498]

11.Tambwekar P and Gombolay M (2024) Towards reconciling usability and usefulness of policy explanations for sequential decision-making systems. *Front. Robot.* AI 11:1375490. [DOI: 10.3389/frobt.2024.1375490]

12.Rizvi, S. S. R., Khan, M. A., Abbas, S., Anwer, N., & Fatima, A. (2022). Deep extreme learning machine-based optical character recognition system for Nastalique Urdu-like script languages. *The Computer Journal, 65*(2), 331-44.[DOI: 10.1093/comjnl/bxaa042]

13.Lee H, Kwon J, Chong NY, Yang W. 2026. Explainable and transferable deep reinforcement learning for adaptive patrol of rail-guided robot system. *PeerJ Computer Science* 12:e3722 [DOI: 10.7717/peerj-cs.3722]

14.Glanois, C., Weng, P., Zimmer, M., et al., (2024). A survey on interpretable reinforcement learning. *Machine Learning, 113*, 5847-90. [DOI: 10.1007/s10994-024-06543-w]

15.Matloob, F., Aftab, S., Ahmad, M., Khan, M. A., Fatima, A., & Sabri, N. (2021). Software defect prediction using supervised machine learning techniques: A systematic literature review. *Intelligent Automation & Soft Computing,* 29(2), 403-21. [DOI: 10.32604/iasc.2021.017562]

16.Li, P., Siddique, U., & Cao, Y. (2025). From explainability to interpretability: Interpretable policies in reinforcement learning via model explanation (arXiv:2501.09858). *arXiv*. [DOI: 10.48550/arXiv.2501.09858]

17.Shi, W., Huang, G., Song, S., Wang, Z., Lin, T., & Wu, C. (2021). Self-supervised discovering of interpretable features for reinforcement learning (arXiv:2003.07069). *arXiv*. [DOI: 10.48550/arXiv.2003.07069]

18.Ahmad, M., Alfayad, M., Aftab, S., et al., (2021). Data and machine learning fusion architecture for cardiovascular disease prediction. *Computers, Materials & Continua*. [DOI: 10.32604/cmc.2021.019013]

19.https://www.kaggle.com/datasets/ziya07/robot-interaction-and-network-performance-dataset