



Research Article

Human-in-the-loop transfer learning in collision avoidance of autonomous robots

Minako Oriyama^{a,*}, Pitoyo Hartono^b, Hideyuki Sawada^c^a Department of Pure and Applied Physics, Graduate School of Advanced Science and Engineering, Waseda University, Tokyo 169-8555, Japan^b School of Engineering, Chukyo University, Aichi 466-8666, Japan^c Department of Applied Physics, Faculty of Science and Engineering, Waseda University, Tokyo 169-8555, Japan

ARTICLE INFO

Article history:

Received 19 November 2024

Revised 26 December 2024

Accepted 17 January 2025

Available online 28 January 2025

Keywords:

Human-in-the-loop

Transfer learning

Autonomous robots

Neural networks

Reinforcement learning

ABSTRACT

Neural networks have demonstrated exceptional performance across a range of applications. Yet, their training often demands substantial time and data resources, presenting a challenge for autonomous robots operating in real-world environments where real-time learning is difficult. To mitigate this constraint, we propose a novel human-in-the-loop framework that harnesses human expertise to mitigate the learning challenges of autonomous robots. Our approach centers on directly incorporating human knowledge and insights into the robot's learning pipeline. The proposed framework incorporates a mechanism for autonomous learning from the environment via reinforcement learning, utilizing a pre-trained model that encapsulates human knowledge as its foundation. By integrating human-provided knowledge and evaluation, we aim to bridge the division between human intuition and machine learning capabilities. Through a series of collision avoidance experiments, we validated that incorporating human knowledge significantly improves both learning efficiency and generalization capabilities. This collaborative learning paradigm enables robots to utilize human common sense and domain-specific expertise, resulting in faster convergence and better performance in complex environments. This research contributes to the development of more efficient and adaptable autonomous robots and seeks to analyze how humans can effectively participate in robot learning and the effects of such participation, illuminating the intricate interplay between human cognition and artificial intelligence.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The integration of artificial intelligence (AI) into robotics has significantly transformed the capabilities and applications of robotic systems. Despite remarkable advancements, developing robots that can autonomously operate in complex and dynamic environments remains challenging. A promising approach to address this challenge is Human-in-the-Loop (HITL) AI.

Recently, HITL AI [1,2] has garnered considerable attention. HITL AI involves the inclusion of human intervention and feedback throughout the machine learning process, from data collection and preprocessing to model training, operational monitoring, and continuous updates. This interactive process between AI and humans aims to leverage human expertise to mitigate the learning difficulties of AI and ensure accountability in AI decision-making. Research employing HITL AI is extensive. For instance, one study [3] proposes a novel role for HITL, where human knowledge is utilized to initialize neural networks rather than merely annotating data or intervening in the learning process. This enables the learning of knowledge that is difficult to

structure as data. Another study [4] demonstrates that incorporating human instructions into training approaches significantly improves learning efficiency in robots' skill acquisition compared to autonomous learning. This past study proposes a simulation platform for surgical robot learning that leverages HITL methods [5]. It demonstrates how integrating human demonstrations into reinforcement learning enhances learning efficiency and increases the success rate of task automation.

The realization of HITL AI allows robots to integrate human common sense and intuitive judgment, enabling them to adapt to new tasks and environments by leveraging human hints and guidance. This process is more flexible than purely data-driven approaches and better suited to handling unexpected situations.

The integration of AI into robotics has seen significant advancements in recent years. Autonomous mobile robots can be realized using reinforcement learning (RL) algorithms without human involvement [6,7]. This study [8] notably demonstrated the successful application of RL to learn control policies for real robots, which were then used in model predictive control (MPC) to achieve tasks in real-world experiments. Additionally, a study on mobile robot path planning [9] proposed RL-based methods for efficient navigation in dynamic environments. However, the

* Corresponding author.

E-mail address: minako.oriyama@suou.waseda.jp (M. Oriyama).

technique either lacks generality or is highly task-specific, making the development of accurate and broadly applicable dynamic human behavior models a significant challenge. Incorporating human common sense into robots is effective in overcoming this challenge. One study [10] proposes a novel approach that combines reinforcement learning with human knowledge. Using a two-stage reinforcement learning model, the first stage involves basic robotic behavior, while the second stage incorporates human common sense as incentive rewards. This enables robots to acquire the necessary skills for task resolution more efficiently during the initial learning stages. Another study [11] suggests a method that enhances the effectiveness of reinforcement learning by providing robots with human common sense knowledge for collision avoidance behavior. Using topological internal representations makes the learning process more intuitive and understandable, facilitating easier comprehension of the robot's decision-making process by humans.

Knowledge transfer is a crucial approach to skill learning in robots. Generally, transfer learning [12,13] leverages knowledge accumulated from one task to improve learning accuracy and efficiency in related but different tasks. A neural network can capture important features during the learning process in the original domain, making it beneficial to transfer parts of the network to another network for learning in a similar domain. In a study involving real robots, a system was developed to learn human preferences from demonstrations of short, standardized tasks and predict user behavior in real assembly tasks [14]. The proposed system uses preference models learned from standardized tasks as prior knowledge and updates the models online, thereby improving the accuracy of human behavior prediction. Additionally, research has explored leveraging experience gained with real robots to accelerate the learning process for new tasks and robots [15].

When transferring knowledge from multiple robots, a method has been proposed to assess the similarity in dynamics between the source and target robots to determine which robot's knowledge is most effective for transfer. In practical robot applications, it is common to transfer neural networks pre-trained in simulations to robots for learning in physical environments (sim-to-real) [16,17]. A sim-to-real approach using progressive networks was proposed to transfer policies learned in simulation to real-world robot arm control tasks [18]. Additionally, deep reinforcement learning has been employed to train mobile robots in navigation tasks within simulated environments that mimic the real world [19]. A deep actor-critic reinforcement learning method was proposed that incorporates human common-sense knowledge into the reward function to improve the efficiency of robot navigation in multi-corridor environments. However, in complex environments, the learning process in simulators can be excessively long and require vast amounts of data, limiting generalization.

Recent studies have explored direct learning in the real world, eliminating the need for simulators [20,21]. For example, one study demonstrated a robot learning to walk on two legs within 20 min across various indoor and outdoor terrains, marking a significant departure from traditional simulation-dependent approaches [22]. Similarly, another study employed the Dreamer algorithm to enable a robot to complete learning directly in the real world without simulation [23]. While these methods succeed in reducing human intervention, it remains important to explore various novel ideas on human involvement into the learning process of robots.

This paper proposes a mechanism for transferring human common sense to robots. This current work comprises two stages of learning: in the first stage, a human transfer his/her knowledge by annotating data or building an intuitive algorithm for

generating teacher signals. This first stage is followed by reinforcement learning of an autonomous robot utilizing a neural network that was initialized by the first stage. The idea is to seed the neural network with human common sense before letting it autonomously learn from the environment. This mechanism allows the combination of human common sense that is not necessarily easy to mathematically express with the strength of reinforcement learning to automatically learn from environments.

The integration of human knowledge into the learning process underscores the importance of the neural network's initial state for subsequent learning outcomes. Prior studies have demonstrated the critical role of initialization in optimizing learning efficiency and effectiveness [24]. For instance, one approach utilized human-defined propositional logic rules to initialize both the weights and structure of a neural network, resulting in enhanced efficiency during the initial search phase [25]. Conversely, suboptimal initialization can increase the likelihood of convergence on local optima, thereby impeding learning progress.

In this research, we emphasize the incorporation of meticulously designed human prior knowledge into the network's initial state. Such prior knowledge is crafted to capture the fundamental structure and rules of the task, ensuring its accuracy and relevance. By embedding this knowledge into the initialization phase, the network achieves stable convergence during the early stages of learning, minimizing unnecessary exploration and enhancing overall learning efficiency.

An additional challenge addressed by this approach is catastrophic forgetting [26,27], which arises when neural networks overwrite parameters shared across tasks during the learning of new tasks. Initializing the network with task-specific, well-curated prior knowledge fosters the development of robust shared feature representations, thereby mitigating parameter corruption. For instance, human-curated features embedded in the initialization process enhance knowledge transferability and resilience, reducing disruptions when adapting to new tasks. This method not only alleviates catastrophic forgetting but also contributes to improved learning efficiency and consistency across varying tasks and environments.

In this study, we focus on the task of learning collision avoidance behaviors [28,29] within a defined physical space. Our proposed neural network builds on prior research [30]. Models learned through deep reinforcement learning [31,32] and path planning [33,34] are typically treated as black boxes, making it challenging to interpret the internal mechanisms and logic behind their decision-making. The proposed approach suggests that using a simpler neural network could facilitate the interpretation of the model's internal mechanisms and logic, aligning with the HITL objective of ensuring accountability in AI decision-making.

This study proposes a HITL AI approach to transfer learning, where human knowledge is imparted to robots. Additionally, the concept of "transfer learning" in this research aligns with prior studies [35]. Traditional transfer learning involves applying parts of a neural network trained in one domain to learning in another domain. In this research, we first employ supervised learning to reflect human general sensory judgment processes related to sensor data in physical environments, followed by utilizing the learned neural network in reinforcement learning processes in real environments. Thus, instead of transferring parts of the neural network structure, we transfer prior knowledge reflecting human common sense.

We conducted seven experiments to demonstrate the robustness, generalization, and effectiveness of our proposed transfer learning method. These experiments assessed the differences in robot behavior with and without pre-training, the influence of pre-training on subsequent reinforcement learning, and the utilization of human-provided knowledge across different physical

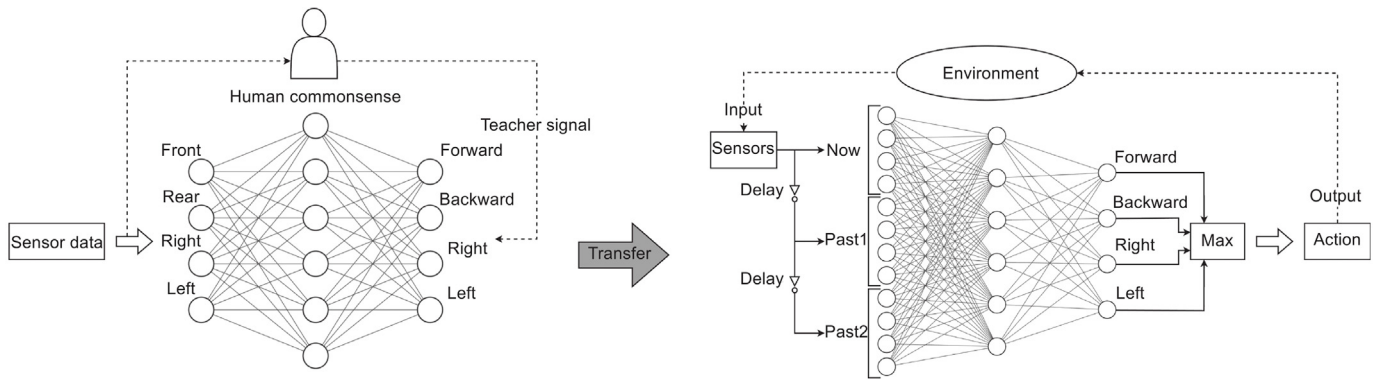


Fig. 1. Overview of the robot's training process.

environments. The validity of our approach was tested using an obstacle avoidance task, showcasing the benefits of human involvement in transfer learning.

The structure of this paper is as follows: Section 2 describes the architecture and dynamics of the neural networks used in this study. Section 3 provides an overview of the robots employed in the research. Section 4 presents and discusses the experimental results of the proposed learning approach. Finally, Section 5 offers a summary and outlines future challenges.

2. Neural network's structure and dynamics

Fig. 1 outlines the robot's training process, which includes human-guided pre-training followed by reinforcement learning. Additionally, the pseudo-code for the entire learning mechanism is summarized in Algorithm 1.

Algorithm 1: Overall learning mechanism

```

1 start
2 robot random walk: data acquirement
3 human annotates sensor data
4 set neural network (input: 4 sensors)
5 until stop criterion
6   input human-annotated data
7   execute offline training
8 end
9 expand neural network (input: 12 sensors)
10 transfer parameters from 4-sensor network
11 randomly initialize remaining weights
12 until stop criterion
13 robot interacts with environment autonomously
   (12 sensors data)
14 execute reinforcement learning
15 end
16 end

```

The elaboration of the overall training process is elaborated as follows.

2.1. Human guided pre-training

The robot is initially provided with prior knowledge through a pre-training phase. In the experimental environment, first, the robot starts at the center of the environment and performs 100 random actions (forward, backward, left, right). For each action, the values of the proximity sensors (front, rear, right, left) were recorded. In this stage, the robot does not execute any learning, but only gathers sensory data to be annotated by humans with a fixed algorithm that reflects the humans' common sense.

The neural network used for pre-training is a fully connected architecture with 4 input nodes, 30 hidden layer nodes, and 4 output nodes. The input layer receives ultrasonic sensor values from the front, rear, right, and left sides, which are normalized and processed as \mathbb{R}^4 vectors. The hidden layer consists of a single fully connected layer, utilizing a sigmoid activation function to introduce nonlinearity. The output layer corresponds to actions for each direction (forward, backward, right turn, left turn). The output values are evaluated using a loss function against the teacher vector. Mean squared error (MSE) is employed as the loss function, and the model parameters are updated through stochastic gradient descent (SGD).

The pre-training neural network shares the same structure as the reinforcement learning network (detailed later). The distinction lies in the input layer: pre-training uses the values of four proximity sensors as inputs, while in the reinforcement learning stage, the inputs are expanded to twelve where the additional eight values are for accommodating the delayed sensory values for the past robot's position. The objective for this expansion is to enable the robot to generate time-mediated strategy rather than an instantaneous reaction. This expansion is necessary for dealing with dynamics environments. In the pre-training phase, the robot relies on human common sense to guide its actions based on this limited set of sensor inputs. Using only four sensors at this stage is intentional, aligning with the human ability to process simple data. Humans can easily interpret and label these basic sensor readings to guide the robot. However, as the number of sensors increases, interpreting the data becomes exponentially more complex, making it difficult for human common sense alone to manage. This highlights the need to limit sensor inputs during pre-training to a manageable level for human annotation. By starting with simpler, easily interpretable data, the robot develops an initial framework for decision-making, preparing it for more complex learning in later stages, where human guidance is less feasible due to the higher volume and complexity of sensor inputs. In the reinforcement learning (RL) stage, the robot interacts with its environment and learns from a larger set of inputs—twelve sensors in this case—allowing it to make more nuanced decisions based on real-time feedback.

In the pre-training stage, the teacher signal is a fixed algorithm based on human common sense that is equivalent with humans' annotation. This HITL approach is essential because humans provide structured, rule-based feedback that helps the robot build a basic understanding. The importance of HITL becomes more evident during the transition from pre-training to the RL stage. While RL allows the robot to learn more complex input-output relationships by interacting with the environment, this process is significantly faster and more efficient because the robot already possesses a basic understanding from pre-training. The combination of human-guided pre-training and autonomous reinforcement learning underscores the importance of HITL in achieving

efficient and effective learning in robotic systems.

As the task is obstacle avoidance, this “common sense” translates to strategies for the robot to move away from obstacles. However, these rules can be subjective and influenced by human experience and preferences. We hypothesize that initializing the network with human common sense enhances subsequent reinforcement learning efficiency and allows human intervention in judgment and control, showcasing the system’s versatility. In this paper, the human-guided fixed evaluation algorithm is shown in Algorithm 2.

Algorithm 2: Human-guided evaluation algorithm

```

1 start
2 load sensor data from robot random walk
3 identify sensor with minimum value
4 initialize variables for storing sensor data and teacher
  signals
5 for each row in sensor data
6   if minimum value is from 'front' sensor
7     set teacher signal to 'back'
8   elif minimum value is from 'back' sensor
9     set teacher signal to 'forward'
10  elif minimum value is from 'right' sensor
11    set teacher signal to 'left'
12  elif minimum value is from 'left' sensor
13    set teacher signal to 'right'
14  store sensor data and teacher signal
15 until stop criterion
16 pretrain neural network with stored data
17 save trained network
18 end

```

2.2. Neural network transfer learning

After the pre-training stage, a transfer of the neural network occurs. Trained network weights and biases are transferred as initial parameters for the reinforcement learning network. However, structural differences necessitate adjustments: the pre-trained network has a four-node input layer, while the reinforcement learning network has twelve. The increase of the inputs is not due to the increase in the number of sensors but the utilization of the delayed values of the sensors to accommodate the time dynamics in inputs. Specifically, the weight matrix of the first linear layer in the pre-trained network overwrites the first four columns of the corresponding matrix in the reinforcement learning network. The remaining columns are randomly initialized. For other layers (hidden and output), weights and biases are transferred directly due to matching input dimensions.

2.3. Real-time reinforcement learning

After pre-training, the network is employed for real-time reinforcement learning.

The reinforcement learning network differs from the pre-trained one (Fig. 1, right): it has twelve input nodes for sensor values and four output nodes for actions, mirroring the pre-trained network. Notably, this study uses time-series sensor values. To adapt to dynamic environments, we employed 12 input nodes representing both sequential and time-series inputs. The first four nodes represent current time t , the next four represent $t - 1$, and the last four represent $t - 2$. This allows the network to consider past values, potentially improving performance in dynamic environments.

Regarding network transfer with differing input nodes, pre-trained parameters are used for the first four input neurons

during initialization. For $t - 1$ and $t - 2$ values at $t = 1$ and $t - 2$ values at $t = 2$, input is set to zero.

The neural network architecture employed in this study mirrors the approach used in previous research [35]. Sensory data is processed to determine contextually appropriate actions. Typically, humans decide their next action based on past data. However, humans cannot explain how they can relate past information to actions. Previous research [35] outputs actions based on real-time input (4 sensor values) and prior knowledge taught by humans. In contrast, our proposed method uses time-series input values, allowing AI to take on the parts that humans cannot compensate for. By clarifying the roles of humans and AI in this way, we can emphasize the HITL nature of our approach.

Initially, sensory inputs are weighted based on their relative importance. These weighted inputs are then integrated with a dynamic bias term, which adjusts based on the current environmental context. The resulting signal is passed through a sigmoid activation function, which serves to normalize the output and prepare it for subsequent processing.

This transformed signal is then propagated to the next layer of the network, where the same process is iterated. Ultimately, the action associated with the most activated output neuron is selected for execution.

Next, the robot performs reinforcement learning to judge the accuracy of the executed action based on sensor values. The reward calculation involves three steps: (1) acquiring sensor values, (2) obtaining sensor values at the subsequent time step after an action, and (3) calculating the reward. The evaluation function $U(a(t))$ assesses the distance between the robot and obstacles before and after the action. This distance is calculated based on changes in sensor values, providing an intuitive and effective measure for obstacle avoidance tasks. At time t , let $d_1(t)$ and $d_2(t)$ be the two smallest sensor values among the four acquired values. At time $t + 1$, the sensor values corresponding to the previously selected directions ($d_1(t)$, $d_2(t)$) are recorded. Using these values, the change in sensor readings is calculated based on the evaluation formula. The evaluation function $U(a(t))$ is obtained by calculating the difference between the mean squared differences of these two values before and after the action (Eq. (2)):

$$d_1(t) < d_2(t) < d_3(t) < d_4(t) \quad (1)$$

$$U(a(t)) = \sqrt{\sum_i^2 (d_i(t+1))^2} - \sqrt{\sum_i^2 (d_i(t))^2} \quad (2)$$

A positive $U(a(t))$ indicates the robot moved away from obstacles, considered a “good” action in reinforcement learning. Conversely, a negative $U(a(t))$ indicates movement towards obstacles, a “bad” action.

For good actions, the winning neuron receives a teaching signal $T_w = 1$, and other neurons receive $T_j = 0$ ($j \neq w$). This promotes the same action for similar future inputs. For bad actions, $T_w = 0$ for the winning neuron and $T_j = 1$ for others, suppressing the executed action and promoting other actions for similar inputs. As the ideal action is unknown, this mechanism is considered reinforcement learning, though its implementation mimics supervised learning.

The teaching signal is expressed mathematically as:

$$T_w^{(k)}(t) = \begin{cases} 1, & \text{if } U(a(t)) \geq 0 \text{ and } k = \arg \max_j O_j^{\text{out}}(t) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$T_j^{(k)}(t) = \begin{cases} 1, & \text{if } U(a(t)) < 0 \text{ or } k \neq \arg \max_j O_j^{\text{out}}(t) \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In this study, the reinforcement learning problem is reformulated as a supervised learning task, with the neural network

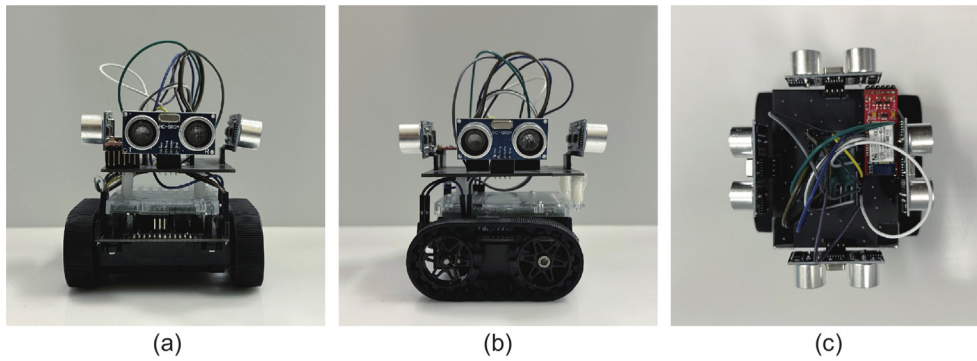


Fig. 2. Outline of the robot. (a) Front view. (b) Side view. (c) Top view.

being trained by minimizing a mean squared error loss (MSELoss) function between the network output and the desired target value.

The network utilizes stochastic gradient descent for optimization, iteratively updating its weights to minimize the loss. Weight updates are determined by the gradient of the error function, which is efficiently computed via backpropagation using the chain rule.

To ensure stable and efficient learning, the weight update equation incorporates a learning rate hyperparameter, which controls the magnitude of each update step, and an exponential decay factor, which gradually reduces the influence of past gradients over time.

The proposed reinforcement learning algorithm employs a neural network to approximate the policy function for state-action mapping. Unlike conventional Q-learning, which relies on a Q-table to estimate the value function, this approach directly maps states to actions through the network. This reduces computational complexity and enhances the system's ability to generalize across similar states, enabling more efficient operation in complex environments where traditional Q-tables face scalability limitations.

Algorithm 3: Reinforcement Learning

```

1 start learning
2 while not converged do
3   acquire current sensor values  $S_t$ 
4   recall the last two sensor values  $S_{t-1}, S_{t-2}$ 
5   calculate output based on  $S_t, S_{t-1}, S_{t-2}$ 
6   execute action
7   acquire new sensor values  $S_{t+1}$ 
8   evaluate the outcome
9   if evaluation is good then
10    reinforce executed behavior
11    suppress alternative behaviors
12  else if evaluation is bad then
13    suppress executed behavior
14    reinforce alternative behaviors
15  end if
16 end while
17 end learning

```

3. Robot platform

In this experiment, we designed and used an omnidirectional robot, shown in Fig. 2. This section details the robot platform, including its chassis, ultrasonic sensors, Bluetooth module, and other hardware components, as well as its motor control and behaviors.

Table 1

Robot specification.

Parameter	Value
Gross weight (g)	262
Dimensions L × W × H (mm)	87 × 98 × 90
Number of proximity Sensors	4
Measurement range (cm)	2–400
Operating frequency (kHz)	40

3.1. Robot platform

Our robot platform is based on the “Zumo Robot for Arduino”, which is designed for tracked movement. The Zumo robot’s tracks enable stable operation on both slippery surfaces and rough terrain. The robot’s design features a two-layer structure: the lower layer houses the Arduino, and the upper layer uses a rigid board. Previous studies [35] used a breadboard with many jumper wires, leading to issues like wiring congestion, poor contact, low durability, and low reproducibility. Using a rigid board instead of a breadboard improved connection stability, space utilization, and wiring visibility.

Four ultrasonic proximity sensors were installed on the front, back, right, and left sides of the rigid board. These sensors measure distances to obstacles by emitting a 40 kHz ultrasonic wave triggered by a 10 μ s pulse. The time elapsed between the trigger and the returning echo is utilized to calculate distance, with a measurement range of 2 cm to 400 cm. Measurements outside this range may be subject to error. Table 1 details the specifications of the robot and sensors.

The Bluetooth module on the upper layer connects the Arduino to a PC via serial communication. Sensor data is sent from the Arduino to the PC via Bluetooth, where a neural network processes the data. The network calculates the robot’s actions and sends them back to the Arduino.

3.2. Actions

We controlled the Zumo robot to perform four movements: forward, backward, right turn, and left turn. The motor speed is controlled using PWM (Pulse Width Modulation), with speeds indicated by the duty cycle of the PWM signal, ranging from -400 to 400 . Negative speeds indicate a reverse direction, so the speed’s absolute value is used, with the direction reversed. The Output Compare Register (OCR_n) is utilized to set the count value at which the PWM signal transitions to a HIGH state. Conversely, the Input Capture Register (ICR_n) is used to define the count value corresponding to the period of the PWM signal.

$$DutyCycle = \frac{OCR_n}{ICR_n} \quad (5)$$

In this experiment, the left and right motors were controlled independently, with a consistent speed of 200 primarily achieved using values of $OCR_1 = 200$ and $ICR_1 = 400$. To execute turns, one motor's speed was set to 0, enabling the robot to pivot around the stationary motor. This setup allows the robot to maneuver flexibly and respond to the neural network's instructions.

4. Experiments

To validate the efficacy of the proposed HITL AI-based transfer learning approach, we conducted six experiments. Each experiment's results contribute to the verification of transfer learning through different training methods and learning stages, as detailed below.

4.1. Pre-training experiments (Experiment A)

The neural network was pre-trained using sensor data and human-generated teacher signals via standard backpropagation. The experimental environment is shown in Fig. 3(a). To create a controlled experimental environment, 12 polypropylene panels were assembled. All experiments started from the marked cross at the bottom left of the environment shown in Fig. 3(a).

The dynamics of the loss function during pre-training is shown in Fig. 3(c), and the graph indicates an exponential decline, demonstrating that the neural network effectively learned from the human-generated data.

Subsequently, an offline experiment was conducted to evaluate the accuracy of the pre-training by generating actions based solely on the data. The pre-trained neural network was tested to see if the robot could appropriately perform collision avoidance actions based on the sensor values received in the experimental environment. The robot started from the bottom left of the environment and predicted and generated 50 steps of action. As a result, the robot moved to the center of the environment and exhibited a behavior of stagnation there (Fig. 3(b)). Additionally, Fig. 4 shows a graph plotting the sensor values closest to the obstacle (the minimum value from the four sensors), averaged over 10 time steps. The upward trend indicates that the robot is gradually moving away from the obstacle. The oscillations in the latter part suggest that continuous movement by the robot is unavoidable. This outcome indicated that the robot acquired a basic policy for collision avoidance based on human commonsense knowledge. However, it was hypothesized that action prediction based solely on pre-training would struggle to adapt to complex and dynamic environments and physical changes. This consideration considers the differences between the pre-training environment and the actual test environment. To verify this hypothesis, we plan to demonstrate it by conducting sequential learning incorporating reinforcement learning using the pre-trained neural network.

4.2. Reinforcement learning experiments (Experiment B)

Next, to demonstrate the utility of pre-training, the robot's actions were generated using sequential learning without pre-training. The robot, lacking prior knowledge, performed reinforcement learning for 50 steps. To assess the system's generalizability, the robot was tested with three starting positions: bottom left, bottom center, and bottom right of the environment.

The robot starting from the lower-left moved straight and collided with the wall (Fig. 5(a), top). The robot starting from the bottom center struggled with the initial movement due to the lack of prior knowledge and repeatedly performed the same action in place (Fig. 5(a), middle). The robot starting from the lower-right continuously reversed and eventually hit the wall

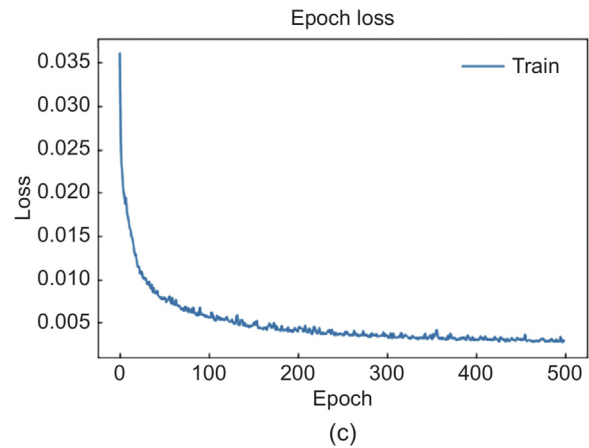
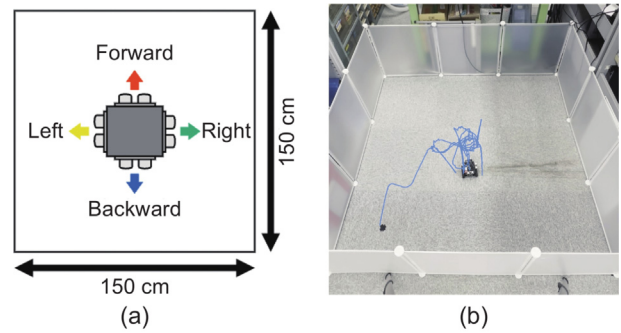


Fig. 3. Pre-training experiments. (a) Experimental environment. (b) Robot trajectory. (c) Training process.

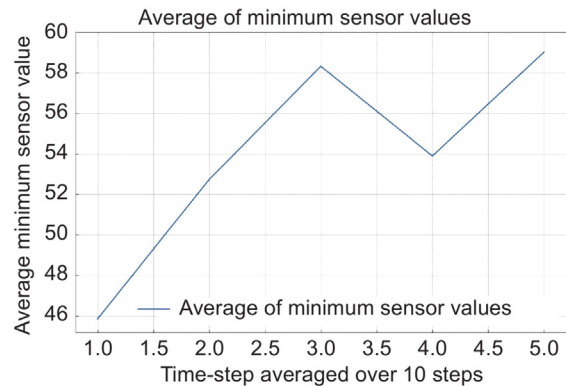


Fig. 4. Average minimum sensory inputs in pre-training experiments.

(Fig. 5(a), bottom). Fig. 6 shows a graph of the sensor value closest to the obstacle, averaged over 10 time steps. Each graph corresponds to the robot's different starting positions. In the case without pre-training, the average sensor value is lower, suggesting that the robot comes closer to the obstacle at some point. These results illustrate the difficulty of learning when the robot has no prior knowledge and must learn from scratch.

4.3. Experiments with human commonsense (Experiment C)

Having demonstrated the accuracy and utility of pre-training, the next experiment involved reinforcement learning augmented with human commonsense knowledge. As in the reinforcement learning-only experiment, three starting positions were tested. The training was conducted over 50 steps.

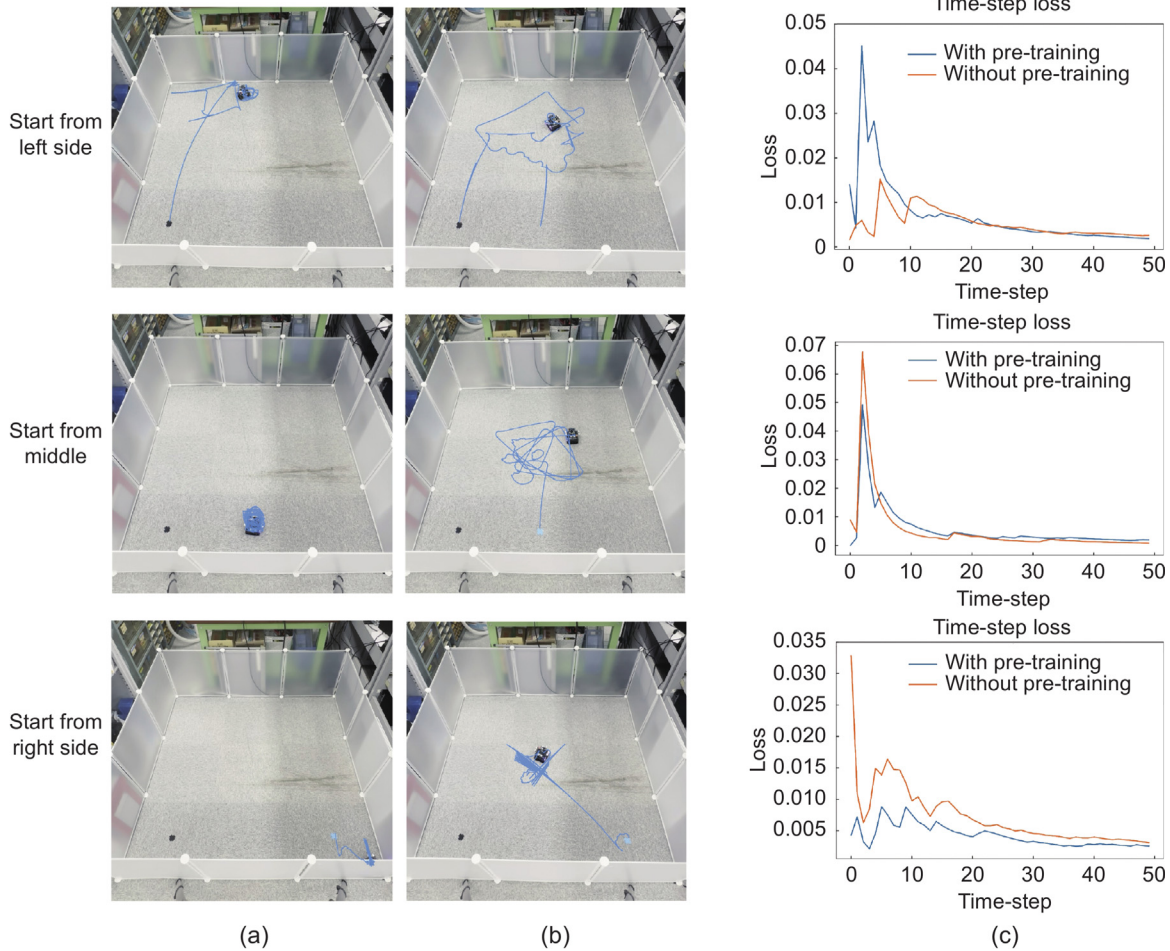


Fig. 5. Reinforcement learning. (a) Robot trajectory without pre-training. (b) Robot trajectory with pre-training. (c) Training process.

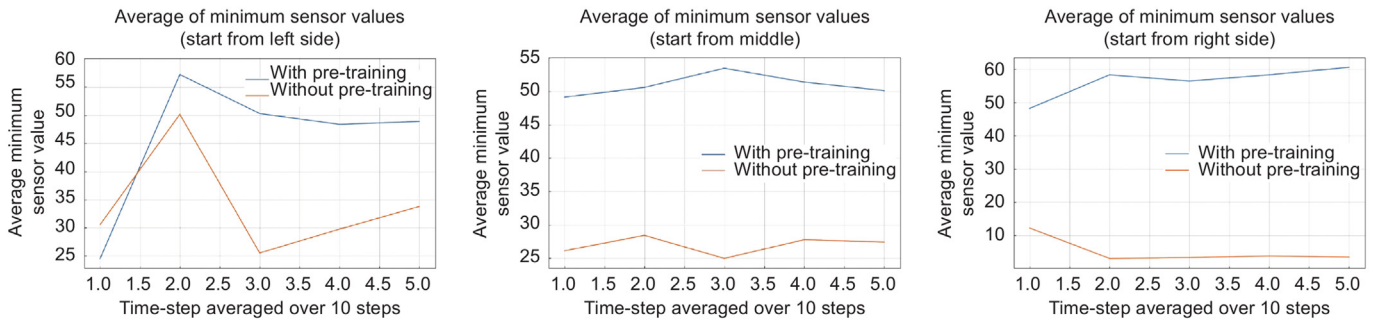


Fig. 6. Average minimum sensor values by starting position.

The robot starting from the lower-left turned directly toward the center of the environment without hesitation, then roamed around the center (Fig. 5(b), top). The robot starting from the bottom center moved straight and then wandered around the center (Fig. 5(b), middle). The robot starting from the lower-right initially changed direction and moved straight to the center, reaching it successfully (Fig. 5(b), bottom). In all cases, after reaching the center, the robot performed back-and-forth movements until the designated time steps were completed. Fig. 5(c) shows the loss function obtained during sequential learning and the loss function obtained in experiment B. This graph illustrates the real-time learning process of the robot, showing a gradual reduction in loss over 50 steps as the motion prediction model improves. The results indicate that the loss decreases over time

regardless of whether pre-training is applied, demonstrating the robot's ability to adapt. The rapid decrease in loss suggests that the model learns efficiently. However, when the robot starts from the right, the loss function fluctuates significantly in the absence of pre-training, indicating instability in the learning process.

These results indicate that with accurate prior information, the robot can learn quickly based on pre-training without the need for repeated trial-and-error collision avoidance. Furthermore, by utilizing past sensor values in addition to current ones, the robot can better understand its position and environment. While both loss functions in Fig. 5(c) show a decreasing trend, the sequential learning with pre-training exhibits lower loss from the initial time steps. This suggests that human prior knowledge allows the robot to establish a certain degree of behavioral policy from

Table 2Welch's t -test: Comparison of results without pre-training vs. with pre-training.

Statistic	Start from left side	Start from middle	Start from right side
t -value	3.41	11.8	28.1
p -value	9.591×10^{-4}	1.411×10^{-20}	9.904×10^{-49}

the outset. As shown in Fig. 6, in the case with pre-training, the average of the minimum sensor values remains stable around 50 cm from the start of the time steps. This suggests that the robot can avoid obstacles early on. The robot learns efficiently by performing real-time reinforcement learning while leveraging previously learned knowledge.

Welch's t -tests were conducted to evaluate statistical differences in the obtained results. The smallest sensor values recorded by the robot at each time step were compared across experiments. The analysis revealed statistically significant differences in all experiments with different starting positions. This outcome can be attributed to the reasonable knowledge provided by humans during pre-training. Since the environment used for reinforcement learning matched the environment in which the robot learned with human-provided knowledge during pre-training, the prior knowledge was effectively utilized in the learning process. These results suggest that significantly improving robot performance requires ensuring that the human-provided knowledge is relevant and applicable to the reinforcement learning experimental environment. Table 2 presents the t -scores and their corresponding p -values for all comparisons.

In the pre-training experiment (Experiment A), the robot acquired a basic policy, which reduced its indecisiveness. In contrast, the robot struggled to find optimal actions in the experiment with reinforcement learning only (Experiment B). Lacking a commonsensical policy, the robot needed to engage in repeated trial and error to discover collision avoidance behaviors. Consequently, in the experiment combining reinforcement learning with pre-training (Experiment C), the robot could perform initial actions based on the acquired basic policy, leading to more refined subsequent action choices.

4.4. Experiment in the dynamical environment (Experiment D)

We created a dynamic environment that changes constantly and conducted experiments to test our hypothesis. Unlike the previous setting where prior knowledge was acquired, this new environment differs significantly despite using the same prior knowledge as in Experiment C.

To create a dynamic environment, we introduced an obstacle robot identical in structure to the learning robot, which moved back and forth along the black line in the image at 70% PWM duty cycle speed (Fig. 7(a)). The learning phase lasted for 50 steps. During this period, the robot navigated from the lower left to the upper part of the environment, approaching the moving obstacle robot, then retreated without collision, moving towards the center (Fig. 7(b)). In contrast, during the experiment without prior knowledge, the robot collided with the moving obstacle robot (Fig. 7(a)). Although it attempted to adjust its trajectory and moved toward the center of the environment, it was unable to implement an effective collision avoidance strategy. The loss function obtained through this sequential learning is depicted in Fig. 7(c). With pre-training, the loss function exhibits low values from the initial stage. In contrast, in the experiment without pre-training, the initial loss is higher. However, as reinforcement learning progresses over time, the learning process eventually converges similarly to the case with pre-training. As in the previous experiment, the plot of the average minimum sensor values is shown in Fig. 8. As shown in Fig. 8, both with and without

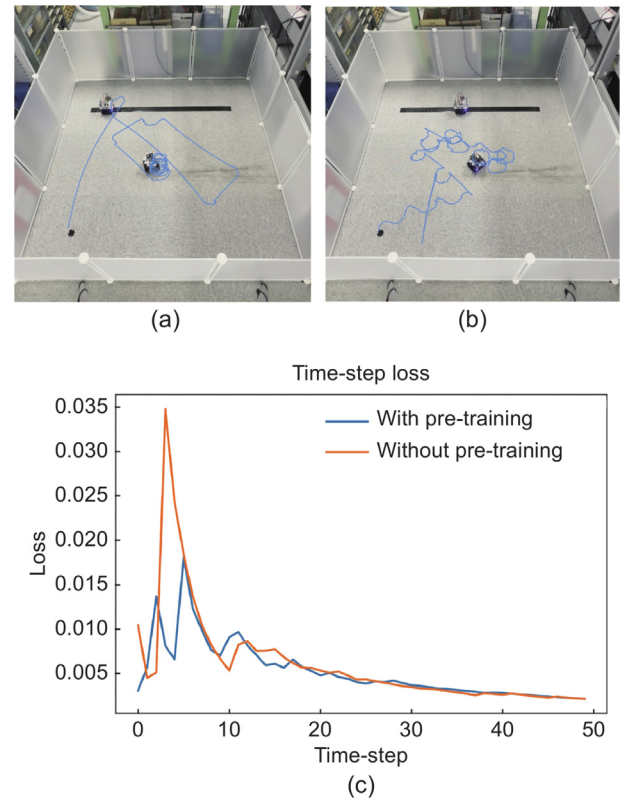


Fig. 7. Learning in the dynamical environment. (a) Robot trajectory without pre-training. (b) Robot trajectory with pre-training. (c) Training process.

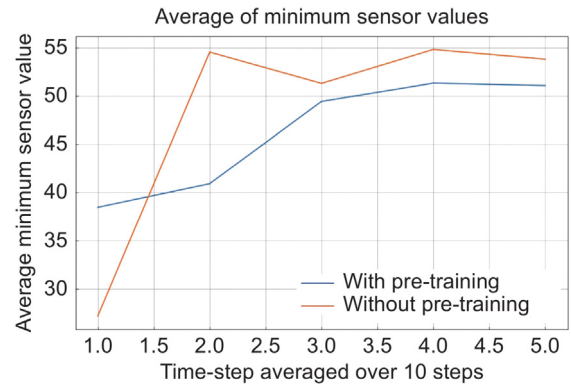


Fig. 8. Average minimum sensor values in dynamical environment.

pre-training, the sensor values eventually stabilize at a consistent level. This indicates that the robot effectively avoids obstacles as the time steps progress. However, with pre-training, the minimum sensor values are higher from an earlier stage, indicating that the robot maintained a greater distance from obstacles. This demonstrates that the robot efficiently used prior knowledge to avoid obstacles, even in a dynamic environment.

The t -test results indicate no statistically significant difference between the two methods, with a t -score of $t = -0.774$ and a p -value of $p = 0.441$, when comparing results without and with pre-training. This outcome may be attributed to the prior knowledge provided by humans not being designed for a dynamic environment. In other words, the relevance of the prior knowledge was limited, leaving the robot in a state similar to starting from random actions, which likely led to the non-significant result.

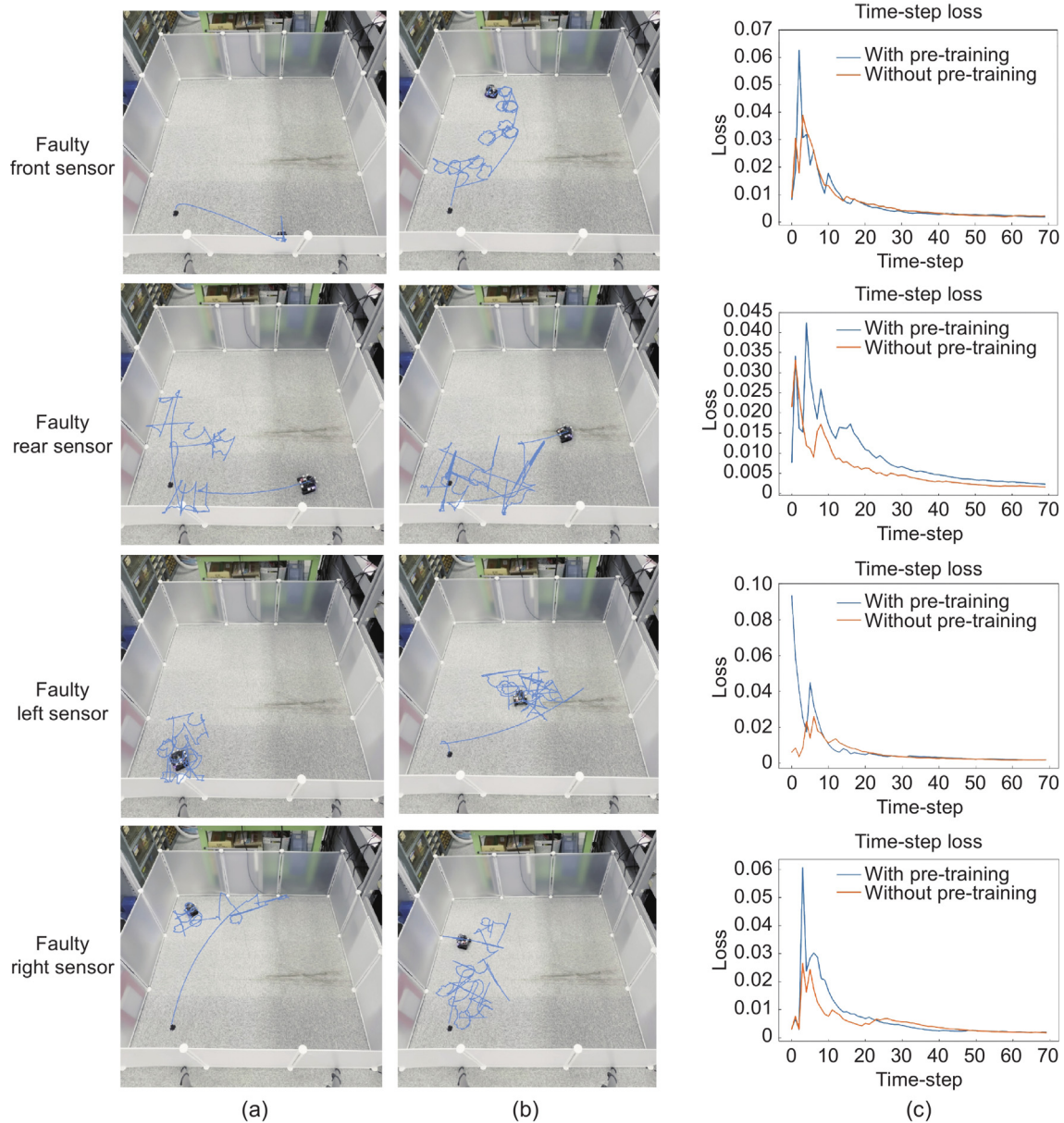


Fig. 9. Learning with faulty input values. (a) Robot trajectory without pre-training. (b) Robot trajectory with pre-training. (c) Training process.

Our findings highlight the effectiveness of using sensor values that account for time series as inputs. Previous research [35] conducted experiments in dynamic environments using only real-time sensor values. In contrast, our study incorporated sensor values from one to two steps earlier, enhancing the robot's spatial awareness. Additionally, our study revealed that the robot could adapt and continue learning in a new environment even when provided with human knowledge from a different setting. By conducting experiments in environments that were constantly changing, the robot showcased its ability to respond effectively to temporal variations in its surroundings, further emphasizing its adaptability and robustness in handling environmental dynamics.

4.5. Experiments with faulty input values (Experiment E)

Next, an experiment was conducted to observe the behavior when physical factors change. In this experiment, one of the four sensors was assumed to be faulty, and the same procedure as in Experiment C was followed. To evaluate the utility of pre-training, we also conducted the experiment without pre-training.

The sensor fault was simulated by adding a random value (-150 to 150) to the sensor readings. The readings from the faulty sensor became random numbers each time, thereby replicating the fault condition. The training was carried out in 70 steps. Fig. 9 illustrates the learning process. Fig. 10 shows a graph of the average values from the sensor closest to the obstacle. When calculating the averages, the values from malfunctioning sensors were excluded. Since malfunctioning sensors produce random values, including them would compromise the reliability of this graph, which aims to trace the robot's movements.

Training with pre-training proceeded without issues, and the robot successfully learned to avoid collisions. Additionally, the weights from the hidden layer to the output layer for actions leading towards the faulty sensor approached zero. This suggests that the input from the faulty sensor contained redundant features, which were compensated for by other sensor inputs and past sensor values. Furthermore, it is likely that during the learning process, the output node responsible for moving towards the faulty sensor was deemed unnecessary, and the network learned to ignore it, causing the associated weights to approach

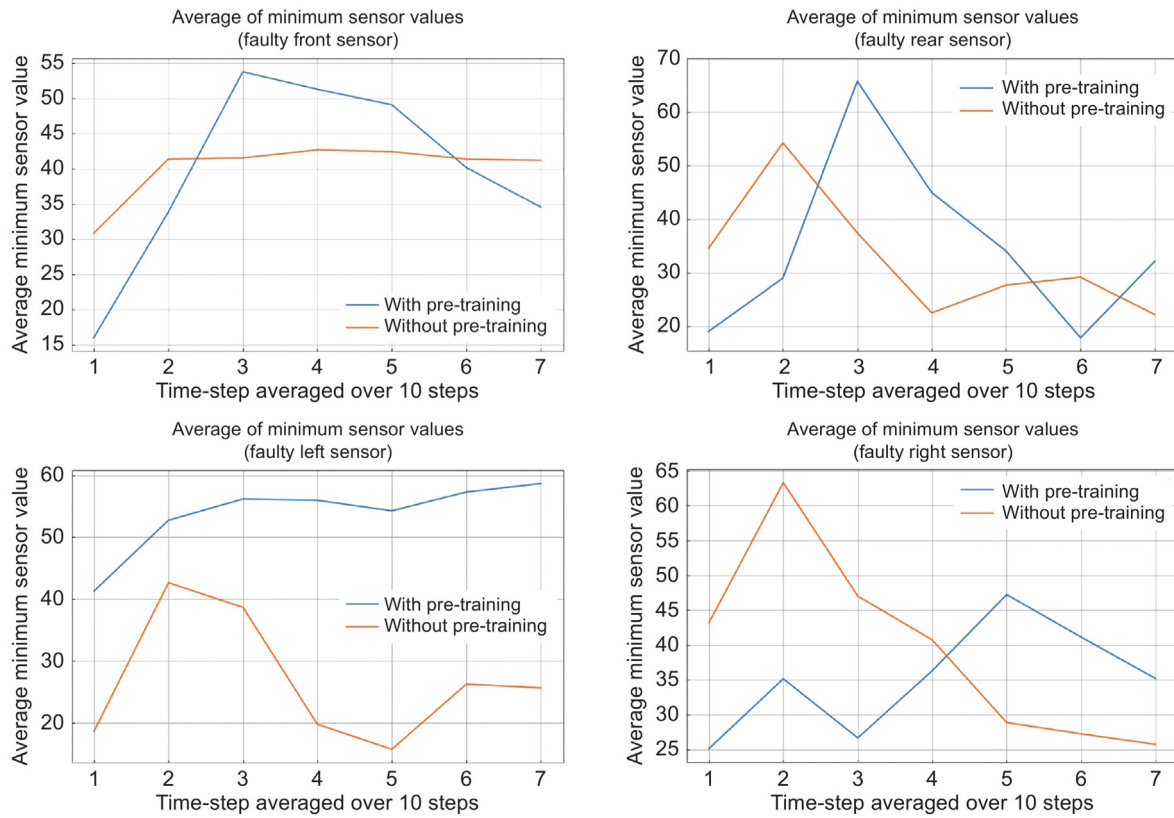


Fig. 10. Average minimum sensor values with faulty sensors: front, rear, right, and left.

zero. In contrast, in experiments conducted without pre-training, the robot collided with the wall at early time steps in every case where four sensors were malfunctioning. These results suggest that the proposed learning mechanism is robust to the physical disturbance of inputs. As shown in Fig. 10, when the front sensor is malfunctioning, the robot struggles with the initial movement even with prior knowledge. This is likely because the malfunctioning sensor causes a mismatch between the robot's prior knowledge of the environment and the current conditions. However, as time steps increase, the robot compensates for the faulty sensor and adjusts its behavior to move away from obstacles. In contrast, without prior knowledge, the initial movements appear random, and while the robot may sometimes avoid obstacles, it often ends up colliding with walls, leading to stable sensor values. The same pattern is observed when the right sensor is malfunctioning. When the rear sensor is malfunctioning, the robot's initial movement is similarly poor, and its behavior remains unstable as time progresses. This instability is likely due to the rear sensor's erratic readings, making it harder for the robot to determine its position compared to when other sensors are malfunctioning. As shown in Fig. 9(c), when the rear sensor is damaged, the loss function decreases more slowly compared to other cases, indicating that the benefits of pre-training are not fully utilized. When the right sensor is malfunctioning, the robot starts from the lower-left, and it can still estimate its position using the remaining functional sensors, making the impact of prior knowledge more significant in this case.

The results of the t -test are presented in Table 3. In this analysis, the minimum sensor values from three functional locations were extracted, excluding the broken sensor values. When the front, rear, and right sensors were faulty, the p -values exceeded 0.05, indicating no statistically significant differences. This outcome is likely because the humans who provided prior knowledge did not account for scenarios where the robot struggled to

Table 3

Welch's t -test: Comparison of results without pre-training vs. with pre-training using faulty input values.

Statistic	Faulty front sensor	Faulty rear sensor	Faulty left sensor	Faulty right sensor
t -value	-0.380	0.194	6.97	-1.95
p -value	0.705	0.846	4.036×10^{-10}	0.053

interpret its environment. However, when the left sensor was damaged, the p -value was very small, revealing a significant difference. This can be attributed to the robot's behavior in the absence of pre-training, where it oscillated to the right and left within the left side of the environment. The damaged left sensor gave the appearance that the robot was successfully adapting to its environment. Although the robot successfully avoided collisions, overall, there were few statistically significant differences. The experiment highlights that the alignment between the reinforcement learning environment and human expectations plays a critical role in achieving significant differences.

4.6. Experiments with faulty output values (Experiment F)

We conducted an experiment to observe how the robot's behavior changes when one of its motor's malfunctions. This was achieved by simulating a fault in one of the two motors, reducing its speed by half. The normal motor speed is 200, but the faulty motor's speed was set to 100. The robot underwent 70 learning steps in this altered state. To assess the utility of pre-training, we also performed the same experiment without it. Fig. 11 displays the robot's movement paths and loss functions when either the left or right motor is faulty.

Despite the motor malfunction, the robot was able to adjust its behavior and move towards the center of the environment. When

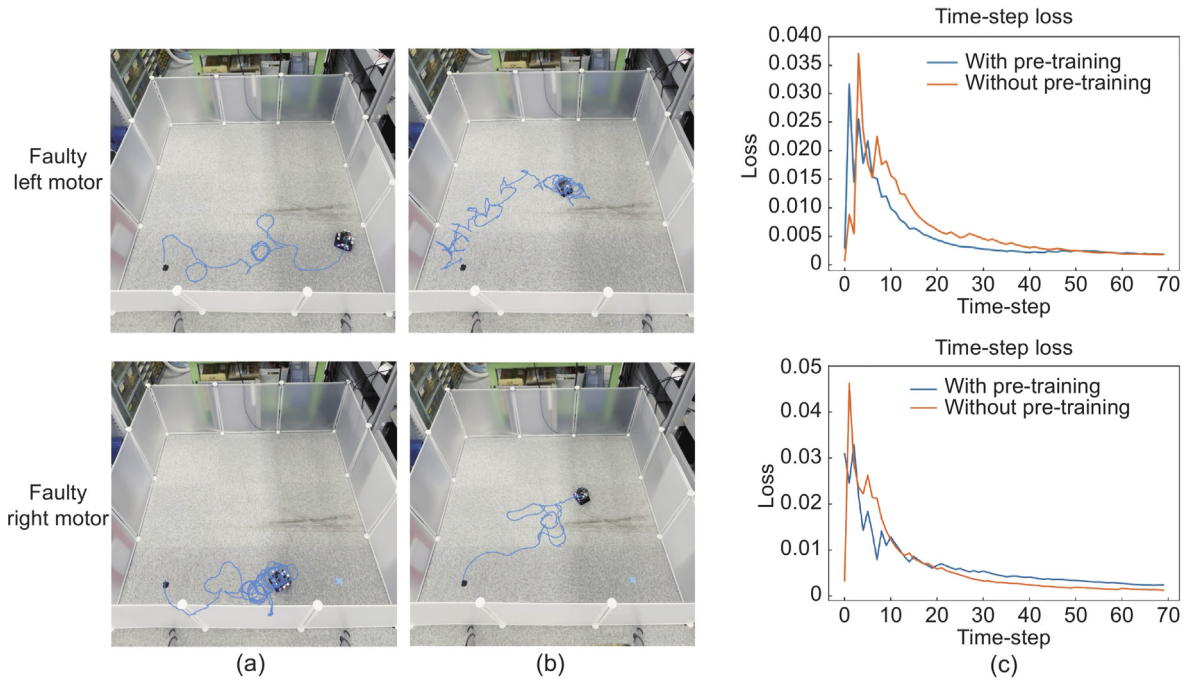


Fig. 11. Learning with faulty output value. (a) Robot trajectory without pre-training. (b) Robot trajectory with pre-training. (c) Training process.

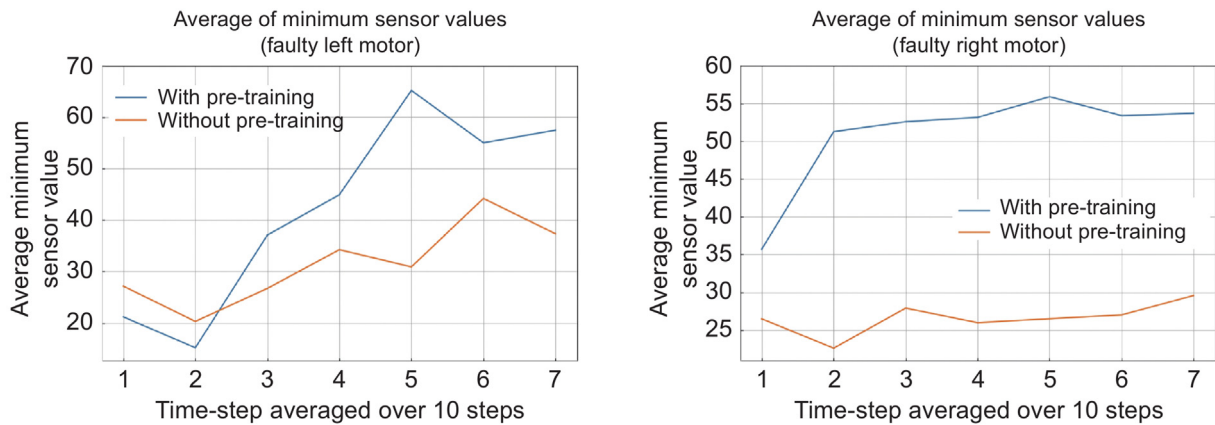


Fig. 12. Average minimum sensor values with faulty motors: right and left.

the left motor malfunctioned, the robot struggled with right turns but managed to avoid obstacles by attempting right rotations or repeatedly adjusting its orientation with left turns. The difference in the reduction rate of the loss function (Fig. 11(c)) between the left and right motor faults is likely due to the robot starting from the left side. With the right motor failure, the robot quickly reached the center by repeatedly moving forward, causing it to make larger right turns. However, with a faulty left motor, the robot had to learn to turn right by manipulating the malfunctioning motor. The influence of the starting position and motor malfunction is reflected in the average minimum sensor value graph (Fig. 12). When starting from the lower-left, the experiment with a malfunctioning right motor, which allowed the robot to automatically move diagonally forward-right, shows that the average minimum sensor value stabilizes earlier.

For cases where the left motor and right motor were broken, the t -tests yielded $t = 3.82 (p = 2.042 \times 10^{-4})$ and $t = 9.29 (p = 2.958 \times 10^{-16})$, respectively, indicating statistically significant differences between the two approaches. This significance can be attributed to the environmental similarities. The environment

used to provide human knowledge was static, and since the experimental environment is also static, the prior knowledge remains applicable. Consequently, the functionality of the motor, whether intact or broken, has minimal impact on the results.

This experiment showed that, just as humans can act appropriately based on their knowledge and experience even when their outputs are compromised, robots can also use prior knowledge to compensate for malfunctions and act accordingly.

4.7. Experiments in continuous control parameters (Experiment G)

We conducted an experiment to give the robot more freedom in its movements. Previous experiments restricted the robot to four directions at a constant speed. In this experiment, we controlled the robot's movements using function values, allowing for greater flexibility.

The control structure involved applying a smooth sigmoid function to the output layer of the reinforcement learning neural network. This function controlled the motor speed, acting as a membership function. We achieved the smoothing by scaling the

sigmoid function with a factor k (Eq. (6)).

$$f(kx) = \frac{1}{1 + e^{-kx}} \quad (6)$$

For this experiment, k was set to 0.5. This normalized the output values between 0 and 1, adjusting the robot's behavior accordingly. The robot's behavior was defined as follows:

Output value 0–0.1 :	stop
Output value 0.1–0.2 :	low
Output value 0.2–0.5 :	moderate
Output value 0.6 and above :	fast

We used the same pre-training data as in Experiment C and conducted training over 50 iterations. The pre-training data involved the robot moving in four specific directions, while the reinforcement learning used this new control mechanism. Figs. 13, 14 and 15 illustrate the robot's trajectory, loss function, and motor output values during this experiment. Fig. 16 presents the graph of the average minimum sensor values.

The robot started well, using collision avoidance strategies from pre-training, and moved toward the center of the environment without hesitation. Fig. 15 shows that the robot increased its motor speed as it approached the center and as learning progressed. This reflects the balance between exploration and exploitation. Fig. 16 clearly shows that the robot develops a collision avoidance strategy early in the time steps and continues to operate while maintaining a consistent distance from obstacles.

In the early exploration phase, the robot did not fully understand the consequences of its actions, requiring various actions at slow speeds to gather information. As learning advanced, the robot shifted to the exploitation phase, where it better understood the environment and the results of its actions. It used successful behavior patterns confidently, increasing motor speed to achieve its goals efficiently.

A t -test produced a t -score of $t = 6.28$ and a p -value of $p = 9.219 \times 10^{-9}$, indicating a statistically significant difference between the conditions with and without pre-training.

In just 50 steps of sequential learning, the robot effectively developed a reliable collision avoidance strategy based on the pre-trained knowledge provided.

5. Conclusion

In this study, we developed a transfer learning process that incorporates HITL dialogue between AI and humans into robots, leveraging human expertise. By pre-training with human commonsense knowledge, we improved reinforcement learning, enabling robots to better handle tasks such as adapting to sensor failures and motor malfunctions.

During the initial training phase, human input provided a dataset of commonsense knowledge about task completion. This human-augmented data allowed the robot to learn and adapt quickly to new environments, significantly reducing the time and resources needed for optimal performance.

As the robot entered the reinforcement learning phase, the benefits of HITL AI became more evident. Using the pre-trained model along with new information acquired by the robot, the exploration and decision-making processes were guided, leading to more informed choices. In obstacle avoidance tests, robots pre-trained with human data navigated more efficiently, showing behavior similar to human attentiveness and foresight. This not only increased the safety and reliability of robot operations but also improved their effectiveness in dynamic and unpredictable environments.

Our research demonstrates that integrating human expertise significantly enhances robot capabilities, improving adaptability,

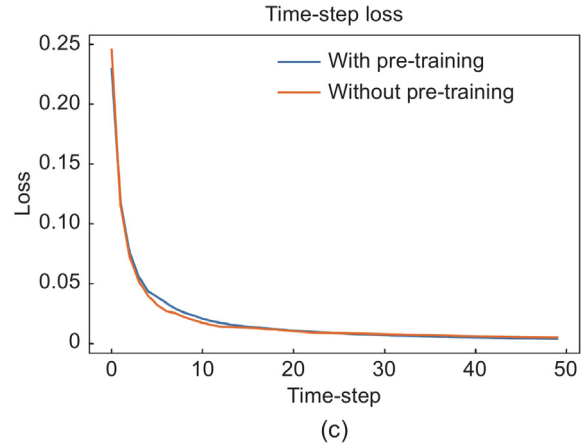
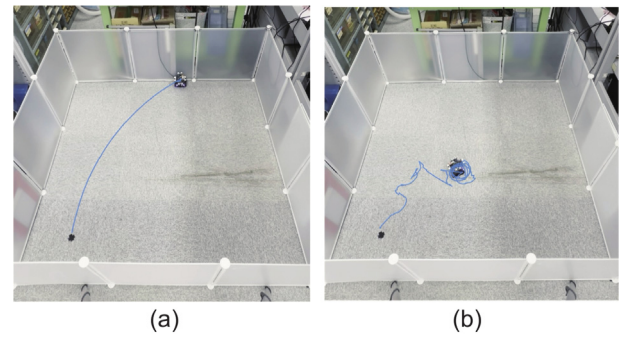


Fig. 13. Learning in continuous control parameters. (a) Robot trajectory without pre-training. (b) Robot trajectory with pre-training. (c) Training process.

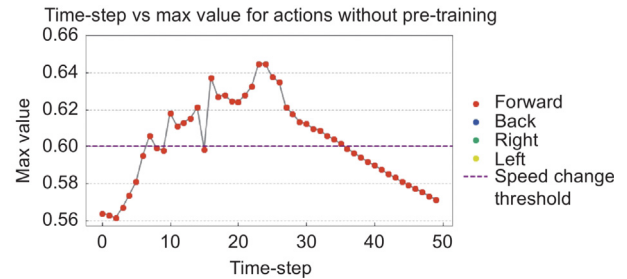


Fig. 14. Transition of output values without pre-training.

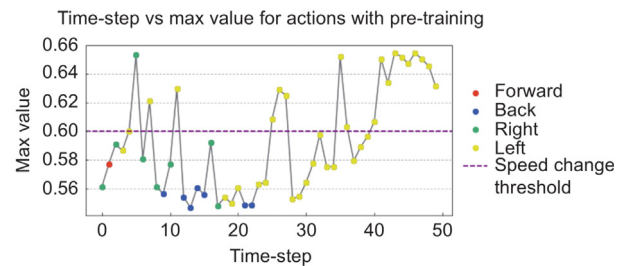


Fig. 15. Transition of output values with pre-training.

efficiency, and transparency. To achieve this outcome, it is crucial that the prior knowledge provided by humans in HITL is not merely “adequate” but also sufficiently comprehensive and tailored to the robot's operating environment and task. If the knowledge is merely adequate, its effectiveness may be limited, preventing the robot from fully realizing its potential. We emphasize the critical importance of aligning human-provided

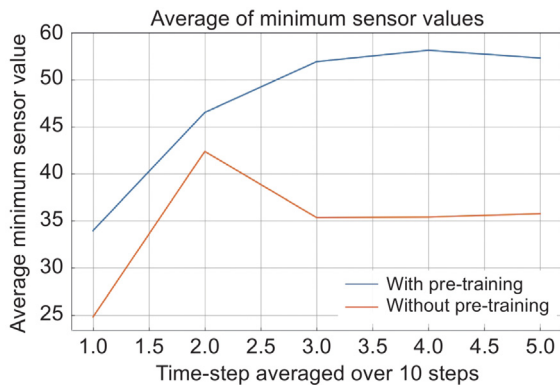


Fig. 16. Average minimum sensor values with robot control by output values.

knowledge with the robot's operational environment in HITL systems. This approach offers a scalable solution for developing robots capable of operating in complex real-world scenarios. As we refine and expand this method, the potential for HITL AI to revolutionize industries and enhance human-robot interaction becomes increasingly clear.

Future work will focus on enabling robots to inherit human intentions and applying HITL AI to a wider range of tasks. Additionally, we plan to conduct user experiments to investigate how different pre-training approaches influence robot behavior.

CRediT authorship contribution statement

Minako Oriyama: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Pitoyo Hartono:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Conceptualization. **Hideyuki Sawada:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Acknowledgment

This research is supported by the research funding of Waseda University, Japan.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.birob.2025.100215>.

References

- [1] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, Ángel Fernández-Leal, Human-in-the-loop machine learning: a state of the art, *Artif. Intell. Rev.* 56 (4) (2022) 3005–3054, <http://dx.doi.org/10.1007/s10462-022-10246-w>, URL <http://dx.doi.org/10.1007/s10462-022-10246-w>.
- [2] Eitan Netzer, Amir B. Geva, Human-in-the-loop active learning via brain computer interface, *Ann. Math. Artif. Intell.* 88 (11–12) (2020) 1191–1205, <http://dx.doi.org/10.1007/s10472-020-09689-0>, URL <http://dx.doi.org/10.1007/s10472-020-09689-0>.
- [3] Issei Suzuki, Pitoyo Hartono, Human-in-the-loop: infusing knowledge into neural networks, in: 2024 IEEE International Conference on Mechatronics and Automation, ICMA, IEEE, 2024, pp. 1665–1670, <http://dx.doi.org/10.1109/icma61710.2024.10633061>, URL <http://dx.doi.org/10.1109/ICMA61710.2024.10633061>.
- [4] Deniz Yilmaz, Barkan Ugurlu, Erhan Oztot, Human-in-the-loop training leads to faster skill acquisition and adaptation in reinforcement learning-based robot control, in: 2024 IEEE 18th International Conference on Advanced Motion Control, AMC, IEEE, 2024, pp. 1–6, <http://dx.doi.org/10.1109/amc58169.2024.10505631>, URL <http://dx.doi.org/10.1109/AMC58169.2024.10505631>.
- [5] Yonghao Long, Wang Wei, Tao Huang, Yuehao Wang, Qi Dou, Human-in-the-loop embodied intelligence with interactive simulation environment for surgical robot learning, *IEEE Robot. Autom. Lett.* 8 (8) (2023) 4441–4448, <http://dx.doi.org/10.1109/lra.2023.3284380>, URL <http://dx.doi.org/10.1109/LRA.2023.3284380>.
- [6] Hee Rak Beom, Hyung Suck Cho, A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning, *IEEE Trans. Syst. Man Cybern.* 25 (3) (1995) 464–477, <http://dx.doi.org/10.1109/21.364859>, URL <http://dx.doi.org/10.1109/21.364859>.
- [7] Guoqian Pan, Yong Xiang, Xiaorui Wang, Zhongquan Yu, Xinzhi Zhou, Research on path planning algorithm of mobile robot based on reinforcement learning, *Soft Comput.* 26 (18) (2022) 8961–8970, <http://dx.doi.org/10.1007/s00500-022-07293-4>, URL <http://dx.doi.org/10.1007/s00500-022-07293-4>.
- [8] Napat Karnchanachari, Miguel I. Valls, David Hoeller, Marco Hutter, Practical reinforcement learning for MPC: Learning from sparse objectives in under an hour on a real robot, 2020, URL <https://arxiv.org/abs/2003.03200>.
- [9] Binyu Wang, Zhe Liu, Qingbiao Li, Amanda Prorok, Mobile robot path planning in dynamic environments through globally guided reinforcement learning, *IEEE Robot. Autom. Lett.* 5 (4) (2020) 6932–6939, <http://dx.doi.org/10.1109/lra.2020.3026638>, URL <http://dx.doi.org/10.1109/LRA.2020.3026638>.
- [10] Siti Sendari, Muladi, Firman Ardiyansyah, Samsul Setumin, Norrima Binti Mokhtar, Hsien-I Lin, Pitoyo Hartono, Common-sensical incentive reward in deep actor-critic reinforcement learning for mobile robot navigation, *Int. J. Innovative Comput. Inf. Control* 20 (2) (2024) 373–389, <http://dx.doi.org/10.24507/ijicic.20.02.373>, URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85186911357&doi=10.24507%2fijicic.20.02.373&partnerID=40&md5=1b41795e03a07d1af74236f5854d811a>.
- [11] Kana Ogawa, Pitoyo Hartono, Infusing common-sensical prior knowledge into topological representations of learning robots, *Artif. Life Robot.* 27 (3) (2022) 576–585, <http://dx.doi.org/10.1007/s10015-022-00776-5>, URL <http://dx.doi.org/10.1007/s10015-022-00776-5>.
- [12] Sinno Jialin Pan, Qiang Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* 22 (10) (2010) 1345–1359, <http://dx.doi.org/10.1109/tkde.2009.191>, URL <http://dx.doi.org/10.1109/TKDE.2009.191>.
- [13] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, Chunfang Liu, A survey on deep transfer learning, in: Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, Ilias Maglogiannis (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2018*, Springer International Publishing, Cham, 2018, pp. 270–279.
- [14] Heramb Nemlekar, Neel Dhanaraj, Angelos Guan, Satyandra K. Gupta, Stefanos Nikolaidis, Transfer learning of human preferences for proactive robot assistance in assembly tasks, in: Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, HRI '23, ACM, 2023, pp. 575–583, <http://dx.doi.org/10.1145/3568162.3576965>, URL <http://dx.doi.org/10.1145/3568162.3576965>.
- [15] Michael J. Sorocky, Siqi Zhou, Angela P. Schoellig, Experience selection using dynamics similarity for efficient multi-source transfer learning between robots, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 2739–2745, <http://dx.doi.org/10.1109/icra40945.2020.9196744>, URL <http://dx.doi.org/10.1109/ICRA40945.2020.9196744>.
- [16] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, Pieter Abbeel, Sim-to-real transfer of robotic control with dynamics randomization, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 3803–3810, <http://dx.doi.org/10.1109/icra.2018.8460528>, URL <http://dx.doi.org/10.1109/ICRA.2018.8460528>.
- [17] Tianhao Zhang, Runyu Tian, Hongqi Yang, Chen Wang, Jinan Sun, Shikun Zhang, Guangming Xie, From simulation to reality: A learning framework for fish-like robots to perform control tasks, *IEEE Trans. Robot.* 38 (6) (2022) 3861–3878, <http://dx.doi.org/10.1109/tro.2022.3181014>, URL <http://dx.doi.org/10.1109/TRO.2022.3181014>.
- [18] Andrei A. Rusu, Mel Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, Raia Hadsell, Sim-to-real robot learning from pixels with progressive nets, 2018, URL <https://arxiv.org/abs/1610.04286>.
- [19] Naijun Liu, Yinghao Cai, Tao Lu, Rui Wang, Shuo Wang, Real-sim-real transfer for real-world robot control policy learning with deep reinforcement learning, *Appl. Sci.* 10 (5) (2020) 1555, <http://dx.doi.org/10.3390/app10051555>, URL <http://dx.doi.org/10.3390/app10051555>.

- [20] Gaoyue Zhou, Liyiming Ke, Siddhartha Srinivasa, Abhinav Gupta, Aravind Rajeswaran, Vikash Kumar, Real world offline reinforcement learning with realistic data source, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 7176–7183, <http://dx.doi.org/10.1109/icra48891.2023.10161474>, URL <http://dx.doi.org/10.1109/ICRA48891.2023.10161474>.
- [21] Charles Sun, Jędrzej Orbik, Coline Manon Devin, Brian H. Yang, Abhishek Gupta, Glen Berseth, Sergey Levine, Fully autonomous real-world reinforcement learning with applications to mobile manipulation, in: Aleksandra Faust, David Hsu, Gerhard Neumann (Eds.), Proceedings of the 5th Conference on Robot Learning, in: Proceedings of Machine Learning Research, vol. 164, PMLR, 2022, pp. 308–319, URL <https://proceedings.mlr.press/v164/sun22a.html>.
- [22] Laura Smith, Ilya Kostrikov, Sergey Levine, A walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning, 2022, URL <https://arxiv.org/abs/2208.07860>.
- [23] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, Pieter Abbeel, DayDreamer: World models for physical robot learning, 2022, URL <https://arxiv.org/abs/2206.14176>.
- [24] R. Andrews, S. Geva, On the effects of initialising a neural network with prior knowledge, in: ICONIP'99. ANZIS'99 & ANNES'99 & ACNN'99. 6th International Conference on Neural Information Processing. Proceedings (Cat. No.99EX378), vol. 1, 1999, pp. 251–256, <http://dx.doi.org/10.1109/ICONIP.1999.843995>.
- [25] Andrew Silva, Matthew Gombolay, Encoding human domain knowledge to warm start reinforcement learning, Proc. AAAI Conf. Artif. Intell. 35 (6) (2021) 5042–5050, <http://dx.doi.org/10.1609/aaai.v35i6.16638>, URL <http://dx.doi.org/10.1609/aaai.v35i6.16638>.
- [26] Robert M. French, Catastrophic forgetting in connectionist networks, Trends in Cognitive Sciences 3 (4) (1999) 128–135, [http://dx.doi.org/10.1016/S1364-6613\(99\)01294-2](http://dx.doi.org/10.1016/S1364-6613(99)01294-2), URL <https://www.sciencedirect.com/science/article/pii/S1364661399012942>.
- [27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, Raia Hadsell, Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526, <http://dx.doi.org/10.1073/pnas.1611835114>, URL <http://dx.doi.org/10.1073/pnas.1611835114>.
- [28] Pinxin Long, Tingxiang Fanl, Xinyi Liao, Wenxi Liu, Hao Zhang, Jia Pan, Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 6252–6259, <http://dx.doi.org/10.1109/icra.2018.8461113>, URL <http://dx.doi.org/10.1109/ICRA.2018.8461113>.
- [29] Bashra Kadhim Oleiwi, Asif Mahfuz, Hubert Roth, Application of fuzzy logic for collision avoidance of mobile robots in dynamic-indoor environments, in: 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques, ICREST, IEEE, 2021, pp. 131–136, <http://dx.doi.org/10.1109/icrest51555.2021.9331072>, URL <http://dx.doi.org/10.1109/ICREST51555.2021.9331072>.
- [30] Pitoyo Hartono, Sachiko Kakita, Fast reinforcement learning for simple physical robots, Memetic Comput. 1 (4) (2009) 305–313, <http://dx.doi.org/10.1007/s12293-009-0015-x>, URL <http://dx.doi.org/10.1007/s12293-009-0015-x>.
- [31] Xiaogang Ruan, Dingqi Ren, Xiaoqing Zhu, Jing Huang, Mobile robot navigation based on deep reinforcement learning, in: 2019 Chinese Control and Decision Conference, CCDC, IEEE, 2019, pp. 6174–6178, <http://dx.doi.org/10.1109/ccdc.2019.8832393>, URL <http://dx.doi.org/10.1109/CCDC.2019.8832393>.
- [32] Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, Jiayu Zhou, Transfer learning in deep reinforcement learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (11) (2023) 13344–13362, <http://dx.doi.org/10.1109/tpami.2023.3292075>, URL <http://dx.doi.org/10.1109/TPAMI.2023.3292075>.
- [33] Karthik Karur, Nitin Sharma, Chinmay Dharmatti, Joshua E. Siegel, A survey of path planning algorithms for mobile robots, Vehicles 3 (3) (2021) 448–468, <http://dx.doi.org/10.3390/vehicles3030027>, URL <http://dx.doi.org/10.3390/vehicles3030027>.
- [34] Bianca Sangiovanni, Gian Paolo Incremona, Marco Piastra, Antonella Ferrara, Self-configuring robot path planning with obstacle avoidance via deep reinforcement learning, IEEE Control Syst. Lett. 5 (2) (2021) 397–402, <http://dx.doi.org/10.1109/lcsys.2020.3002852>, URL <http://dx.doi.org/10.1109/LCSYS.2020.3002852>.
- [35] Minako Oriyama, Pitoyo Hartono, Hideyuki Sawada, Human-guided transfer learning for autonomous robot, in: Neural Information Processing, Springer Nature Singapore, 2023, pp. 186–198, http://dx.doi.org/10.1007/978-981-99-8126-7_15, URL http://dx.doi.org/10.1007/978-981-99-8126-7_15.