ASME Open Journal of Engineering
Online journal at:
https://asmedigitalcollection.asme.org/openengineering

Enhancing Efficiency in Collision Avoidance: A Study on Transfer Reinforcement Learning in Autonomous Ships' Navigation

Xinrui Wang
Department of Aerospace and Mechanical Engineering,
University of Southern California,
3650 McClintock Avenue, OHE 400,
Los Angeles, CA 90089-1453
e-mail: xinruiw@usc.edu

Yan Jin[1]
Department of Aerospace and Mechanical Engineering,
University of Southern California,
3650 McClintock Avenue, OHE 400,
Los Angeles, CA 90089-1453
e-mail: yjin@usc.edu

Collision avoidance in ships and robotic vehicles exemplifies a complex work process that necessitates effective scenario recognition and precise movement decision-making. Machine learning methods addressing such work processes generally involve learning from scratch, which is not only time-consuming but also demands significant computational resources. Transfer learning emerges as a potent strategy to enhance the efficiency of these engineering work processes by harnessing previously acquired knowledge from analogous tasks, thereby streamlining the learning curve for new challenges. This research delves into two critical questions central to optimizing transfer reinforcement learning for the work process of collision avoidance: (1) Which process features can be successfully transferred across varying work processes? (2) What methodologies support the efficient and effective transfer of these features? Our study employs simulation-based experiments in ship collision avoidance to address these questions, chosen for their intrinsic complexity and the varied feature recognition it demands. We investigate and compare two transfer learning techniques—feature extraction and finetuning—utilizing a lightweight convolutional neural network (CNN) model pretrained on a base case of a comparable work process. Pixel-level visual input is leveraged to cover different numbers of encountering ships and fix the input size for the model. This model adeptly demonstrates the feasibility of transferring essential features to newer work process scenarios. Further, to enhance realism and applicability, we introduce a simplified yet comprehensive ship dynamic model that considers the substantial effects of ship inertia, thereby refining the interaction between the model and its environment. The response time is embedded into the reward function design to be considered for policy training. Experimental outcomes underscore the transferability of diverse process features and evaluate the relative effectiveness of the employed transfer methods across different task settings, offering insights that could be extrapolated to other engineering work processes. [DOI: 10.1115/1.4065831]

Keywords: artificial intelligence, deep learning, transfer learning, RL, collision avoidance, artificial intelligence/machine learning, decision-making, knowledge engineering

1 Introduction

In recent years, the integration of artificial intelligence (AI) into engineering work processes has significantly advanced the automation and optimization of complex tasks, such as vehicle collision avoidance and robotic control. Utilizing AI revolutionizes control systems and manufacturing automation, promotes more efficient workflows, enhances decision-making, and leads to superior outcomes. Among AI techniques, reinforcement learning (RL) stands out for its ability to acquire deep insights into work processes by interacting with the environment [1–4]. However, the inherent complexity of most engineering tasks means that beginning each learning process from scratch using RL can be both labor-intensive and computationally demanding [5,6]. Transfer learning has thus gained recognition as an effective approach to enhance learning efficiency. This technique optimizes the learning process by applying knowledge from previously mastered tasks to new, related ones [7–10]. Effective transfer in RL requires a thorough investigation into the transferability of features across different work processes and the development of a tailored training model that facilitates this transfer. Additionally, it is crucial to design a model that considers the features' impacts and the similarities between source and target tasks to ensure successful knowledge transfer.

This study aims to explore the transferability of process features in the context of ship collision avoidance and identify effective methods for transferring these features when RL is used to train maneuvering control systems for collision avoidance at sea. Although RL has been effectively employed in previous ship

[1]Corresponding author.

Manuscript received April 19, 2024; final manuscript received June 18, 2024; published online July 12, 2024. Assoc. Editor: Rouzbeh Amini.

collision avoidance efforts, the complexity of the process often necessitates prolonged training periods, thus posing significant challenges to training efficiency in system development. In our earlier work on RL-based collision avoidance [11], we introduced a belief-based transfer RL method designed to expedite the training process. This method, however, needed to address the transfer of process features directly, thus restricting its applicability to a limited range of scenarios. To our knowledge, feature-level transfer learning has yet to be employed to enhance the efficiency of RL in capturing work process knowledge. Previously, we transferred an entire network, pretrained in a related but more straightforward base case, to a target case [11]. This model was then adjusted through transfer belief—determining the reliability of actions predicted by the base network—and transfer period—specifying the duration for which the base network's guidance is utilized. While this approach has been shown to improve training efficiency in scenarios where the target case closely resembles the base case, its effectiveness diminishes as the similarity between the base and target cases decreases due to the transfer being based on the entire network rather than on specific process features.

Drawing from the deep learning literature on image processing, it is established that the lower layers of a neural network generally process universal features. In comparison, the deeper layers are responsible for capturing more specific features crucial to decision-making [12]. Applying this understanding to RL scenarios for work processes suggests that transferring an entire network might inadvertently include base case-specific features in the deeper layers. These features may be less effective when the target case diverges significantly from the base case. Consequently, our research explores feature-based transfer RL methods, where selected network weight parameters remain frozen while others are finetuned during training for the target case. Our empirical findings from experiments on the ship collision avoidance process affirm the existence and transferability of work process features. Moreover, we have derived a correlation between the success of feature transfer and the similarity between the base and target cases [13].

RL for various work processes often utilizes graphical features from dynamic image frames [14,15]. The ship collision avoidance process exemplifies a particularly complex scenario for large ships with significant inertia navigating through congested waterways [16]. Beyond the geometrical features extracted from images during RL, other factors closely related to the work process play critical roles. For instance, in potential collision scenarios, the risk of collision is typically evaluated using specific risk assessment methods [17,18], and appropriate maneuvers are determined based on these risk levels [19,20]. Additionally, the execution of these maneuvers can experience considerable delays due to the substantial inertia of large ships. Previous research has explored the differences in feature transfer between multiple pairs of base and target cases, revealing variability in the transfer success [11,13]. Typically, human evaluators have subjectively assessed the similarity between these cases, complicating the transfer of feature-based work processes. This study poses two key research questions to address these complexities and improve learning efficiency: (1) *Which features in the ship collision avoidance process are transferable, and how does their transferability vary*? (2) *How does feature transferability evolve throughout the training pipeline in relation to the changing similarities between the base case and target cases*?

To address the first research question, our study incorporates specific work process features into a carefully designed case study, aiming to explore the transferability of these features across different scenarios. We manipulate two critical features from the base case to create two distinct target cases. Each alteration serves to assess how variations in these features affect their transferability. For the second question, we establish a common base case and extract the same set of work process features, applying them to several target cases that exhibit varying degrees of similarity to the base case. This approach allows us to systematically evaluate how feature transferability evolves as the similarity between the base case and target case changes, providing insights into the dynamics of feature-based learning across diverse scenarios.

The remainder of this article is structured as follows: Sec. 2 provides a review of the literature pertinent to this study. Section 3 details the methods applied in our research. Section 4 outlines the design of the case study. The results of these case studies are presented and analyzed in Sec. 5. Finally, Sec. 6 offers conclusions and suggests directions for future research.

2 Related Work

This research employs simulation-based studies of ship collision avoidance to investigate the transferability of work process features, leveraging the inherent complexity and diversity of these scenarios. We incorporate the International Regulations for Preventing Collisions at Sea (COLREG) to encompass various situations and corresponding collision avoidance maneuvers thoroughly. A deep RL approach is adopted to examine the potential for acquiring and reapplying knowledge across different ship collision avoidance scenarios. In the following subsections, we provide a concise review of the literature concerning COLREG rules, reinforcement learning applications in ship collision avoidance, and several pertinent transfer learning methodologies.

2.1 COLREG Rules. To provide more situational awareness and reduce marine collisions, there are "rules of the road" dictated by the International Regulations for Preventing Collisions at Sea (COLREG) [21], by which the own-ship—i.e., the learner ship in our research—will be considered having the "right of the road" or should "give way" to the encountering target-ship—i.e., any other ship that moves based on its planned course and speed—depending on which COLREG region the target-ship is currently in. The COLREG region separation and the COLREG-complaint collision avoidance action are shown in Fig. 1.

When encountering target-ships in the moving process, to avoid a collision, the own-ship is supposed to make maneuver decisions by following the existing guidelines of COLREG. It is worth mentioning that although the regulations are supposed to be followed by all ships, there is no guarantee that a target-ship will follow the rules, especially when different ships can interpret the situation differently. This phenomenon is reflected in the case studies of this research.

2.2 Reinforcement Learning-Based Ship Collision Avoidance. Collision avoidance has been a popular research topic for years in the unmanned surface vehicle and ship navigation domains [22,23]. Researchers have taken deep RL as an essential approach in recent years. Various learning strategies are continuously proposed, enabling the agent to learn collision avoidance policies by interacting with the environment [24–26].

To be applicable, most RL-based ship collision avoidance studies were combined with risk assessment to reduce failures and make avoidance actions compliant with COLREG constraints simultaneously. For instance, Fan et al. incorporated a risk factor into the design of a collision avoidance reward function to penalize actions against the COLREGs based on the associated collision risk. This risk factor was calculated using the distance to the closest point of approach (DCPA) and the time to the closest point of approach (TCPA), which represent the minimum expected distance and time, respectively, at which the own-ship and the target-ship would be nearest to each other—potentially leading to a collision. Additionally, the ship domain was defined as an area within which the own-ship must initiate evasive maneuvers if an obstacle enters [18]. In other research, a more sophisticated ship domain design was adopted, incorporating a mandatory collision avoidance zone where the own-ship could deviate from the COLREGs to avoid emergent collisions. Different artificial

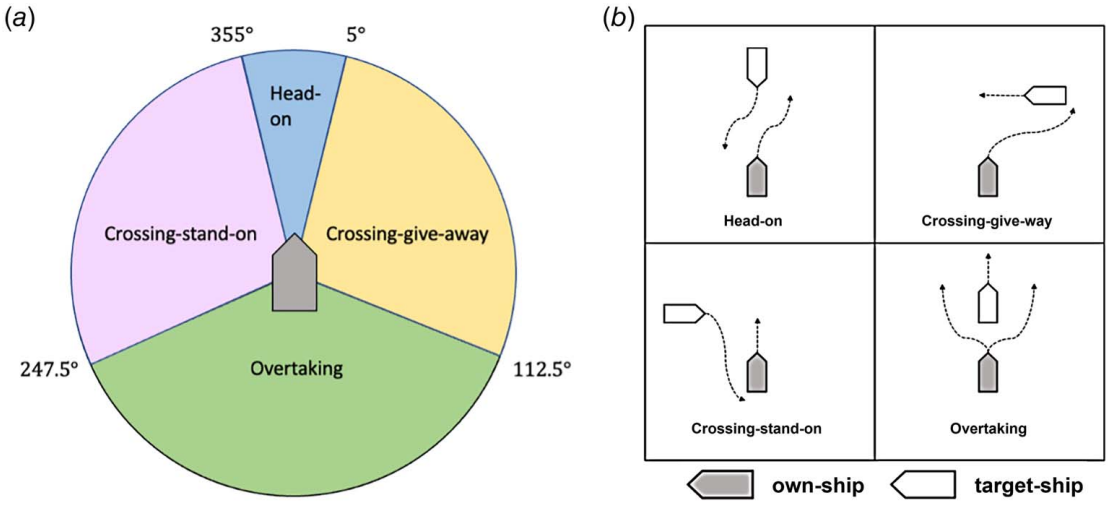

Fig. 1 Illustration of COLREGs: (*a*) COLREG regions and (*b*) COLREG-compliant collision avoidance actions

potential fields were applied across various zones to guide the own-ship toward goals and away from obstacles, thus mitigating collision risks. However, the desired movement direction stipulated by COLREGs, such as rewarding the own-ship more for turning right to give way rather than turning left, was not addressed [20]. Inspired by these studies, our research utilizes a comprehensive ship domain design for risk assessment. COLREG-compliant actions were enforced explicitly through positive or negative rewards. Experimentally, a shaping reward mechanism was employed to provide a stronger signal, as explored in comparative studies of different reward-shaping strategies [27].

Ship dynamics is another big issue with the RL-based approach. Unlike other RL games, an action like a heading or rudder angle changing command could not be completed with a single step. The ship dynamic simulation should be embedded in the training process to address it. Xia et al. utilized a three-degree-of-freedom kinematic model. The output actions were used to calculate the surge force and yaw moment [24]. Sawada et al. used the output command rudder angle to calculate the ship's heading angle and rudder angle based on Nomoto's equation [28] and Runge–Kutta integration [26]. These researchers conducted the simulation separately from the model, leaving the response time uncounted for. However, longer response time causes more time-steps, so more step-penalty should be applied in RL scenarios. In reality, a command with a longer response time is also less preferable due to the higher energy cost. In our research, we have developed a simplified yet comprehensive ship dynamic model that accounts for the significant effects of ship inertia. We also modified the step-reward to include all response time-steps, thereby enhancing the decision-making process by integrating response times directly into the model's calculations.

Lastly, a common challenge in navigation is that an own-ship frequently encounters varying numbers of target-ships. It is crucial to include the status of target-ships in the state input to models as essential information. However, the model input size is fixed, leading to incompatibility between the input size and the varying number of target-ships. To address this issue, Zhao et al. categorized the target-ships into four areas according to COLREGs and configured the state of the target-ship based on these four divisions to fix the input size. Thus, multiple-ship encounters could be handled. However, due to division limitations, the model can observe at most four nearest target-ships [29]. Previous research in our lab utilized images from radar systems as model input; the rich information from an accessible range was covered, and the number of target-ships would not affect the input size [11]. Following this strategy, multiple target-ships within the range of the ship domain are addressed by visual input in this research.

2.3 Feature Extraction and Transfer Learning. Feature extraction is a crucial technique in many recognition-based transfer learning applications, as it aims to extract informative and discriminative features from language, speech, image, or other raw data [10,30]. For image feature extraction, feature refers to the patterns in the given images, e.g., points, edges, and corners, providing critical information to identify associated objects [31] or work for other relevant tasks like 3D reconstruction, semantic segmentation, or image generation [32,33].

Two main categories of image features have been commonly used, hand-crated, and automatic-inferred features [34], competing in different tasks. Human experts explicitly design hand-crafted features based on domain-specific knowledge, which is still widely used despite the emergence of various deep learning models [35]. The bag-of-visual-words model is proposed to represent the overall feature of an image with a histogram of independent features, generating a sparse vector that can be used in classification tasks [30]. In robotics, hand-crafted features are utilized to construct vector input of transfer learning training. For instance, Ahmed et al. [14] designed a hand-crafted observation leveraging heuristic knowledge for transfer reinforcement training of robotic hands. The input vector is constructed with the end-effectors' positions, joint positions, and scene information, providing a rich and descriptive representation of the robotics state. Instead of making an effort to design such feature representations, deep learning has been introduced to automatically extract features from input images through an end-to-end approach, leveraging the power of convolutional networks and transformers. Many well-designed models are published for research and engineering applications [36,37]. Chen et al. employed deep learning techniques to enhance the garbage sorting task by leveraging visual features. Region Proposal Generation and VGG-16 are utilized to construct a machine visual system, completing the object detection and pose estimation tasks. The extracted visual information is then utilized to guide the manipulator in implementing object classification and grabbing [38]. Brohan et al. [39] proposed a pretrained robotic transformer on large-scale real-world robotic tasks, which has been proven capable of absorbing various features in the long-time pretraining process and is generalizable to downstream tasks. In this research, the hand-crafted feature requires expertise in the ship collision avoidance domain, and it may or may not prove itself. Automatic-inferred features are utilized instead. The input image is a 2D gameplay window containing simple geometric shapes, and we define the conceptual representations. The trending models are not chosen due to their large-scale data requirement and generality. A light convolutional neural network (CNN) model has been designed and proved capable of extracting features of the ship collision avoidance process.

In the area of transfer learning, the feature extraction technique plays an important role, pretraining the network in one task to extract essential features and reusing them in the related target task [40]. Another commonly used method is finetuning, which follows the same procedure as feature extraction but re-trains the whole or partial layers of the pretrained network to finetune the model in the target task [12]. Each method has advantages and limitations and can be effective depending on the use case. Furthermore, combining both methods in the same pipeline can provide a more comprehensive solution [33]. The feature extraction method is more efficient. The transferred parameters need not be trained again since useful features have already been extracted from the base case. When the similarity between the base case and target cases is low, it is necessary to use the finetuning method. The copied convolution layers may contain some dataset-specific features because of the big difference between the two datasets; thus, it can work better to be updated during training to fulfill the gap. Feature extraction and finetuning methods are utilized in this research. It is worth comparing the work process feature transferability with these two settings.

3 Methods

We adopt a computational empirical approach to explore the transferability of ship work process features within transfer RL scenarios. This approach includes a simulation environment embedded with ship dynamics, facilitating interactions between the own-ship and its environment, enhancing realism and accuracy. We employ an RL algorithm to facilitate knowledge acquisition and feature identification. Additionally, our transfer learning strategy reuses this acquired feature-based knowledge in varying scenarios, both similar and different. The integration of ship dynamics into the simulation mirrors real-world conditions more closely and plays a pivotal role in maneuvering decisions—critical for successful collision avoidance. The RL algorithm incorporates ship dynamics into the reward function design, which helps in extracting meaningful maneuvering process features. The transfer learning approach enhances training efficiency by leveraging pretrained knowledge and achieves comparable outcomes in reduced training times [11,13]. Furthermore, layer-level analysis of feature transfer helps assess the transferability of different work process features, providing valuable insights for adapting the algorithm to new environments or scenarios.

3.1 Ship Dynamics. This study uses a simplified rudder-controlled ship dynamic model to simulate the motion of the own-ship and target-ships. The ship's state includes angular velocity, course, x and y coordinates in the world frame, and the linear velocity of the ship, described by x_1 to x_5, as shown in the following equations:

$$x_1 = \dot{\varphi} \tag{1}$$

$$x_2 = \varphi \tag{2}$$

$$x_3 = x \tag{3}$$

$$x_4 = y \tag{4}$$

$$x_5 = v \tag{5}$$

The equations of ship dynamics are

$$\dot{x}_1 = -\frac{x_1}{T} + \frac{K}{T}u_1 \tag{6}$$

$$\dot{x}_2 = x_1 \tag{7}$$

$$\dot{x}_3 = x_5 \cos(x_2) \tag{8}$$

$$\dot{x}_4 = x_5 \sin(x_2) \tag{9}$$

$$\dot{x}_5 = a(u_2 - x_5) \tag{10}$$

K is the rudder gain and is set as 0.1555. T is the ship's inertia, which is 73.77. These values come from the settings of a previous MTI project [41]. u_1 and u_2 are the rudder angle and velocity input of the system. The speed x_5 is kept at a constant number of 15 m/s for implementation simplicity. So, the command velocity input u_2 always equals x_5, and acceleration $\dot{x}_5$ is kept at 0 during the training process. x_2 and x_1 represent the heading angle and angular velocity. During training, x_2 is assigned by the selected action of the neural network, as shown in Table 1. Then, the rudder command u_1 is set based on the selected action x_2 following Eq. (11). The own-ship starts steering with this command following the above ship dynamics. The state of the own-ship keeps changing until x_1 and x_2 decay to nearly 0 deg, indicating that the ship completes the steering action, and the heading direction returns to facing forward. The next action will be selected, and the same process will continue until the end of this episode.

When x_2 is relatively small, a linear controller is applied to control the steering process. u_1 is set to be $k \times x_2$, where k is a constant coefficient to guarantee convergence. For the large target course, more steps need to be taken until x_1 and x_2 decay to nearly 0 deg and a primary delay in response to the rudder motion is displayed. To accelerate the process of reaching a command course, a nonlinear component is introduced to handle large rudder angle inputs (greater than φ_1), avoiding an excessively long time for converging:

$$\begin{cases} u_1 = k \times (x_2 + \text{sign}(x_2) \times \varphi_2) & \text{for } |x_2| > \varphi_1 \\ u_1 = k \times (x_2) & \text{for } |x_2| \le \varphi_1 \end{cases} \tag{11}$$

Including the nonlinear function is equivalent to starting with the original command heading course plus an offset course φ_2 in the same direction as the original command. This results in a larger angular acceleration x_1, which shortens the convergence process. Threshold φ_1 to trigger the nonlinear acceleration is set at 10 deg, and the extra command course φ_2 is set to 70 deg. These values were determined through experiments and struck a good balance between convergence rate and avoiding overshoot.

3.2 Reinforcement Learning Training Mechanism. The own-ship is trained as a learning agent. It would teach decision-making strategies to reach the waypoint and avoid collision with target-ships. The target-ships are assigned a fixed starting point, destination, moving speed, and direction. A deep Q network with experience replay is used in this study to train the agent to learn the crucial features in the work process [2]. Unlike some policy-based algorithms (such as PPO [4] and SAC [42]), which have separate policy and evaluation networks, the deep Q network is value based and constructed as an end-to-end structure. This makes using a transfer learning approach more straightforward, as pretrained parameters can be directly loaded into the Q network.

Table 1 Agent action space

Action	ω(deg)
a_1	−25
a_2	−20
a_3	−15
a_4	0
a_5	15
a_6	20
a_7	25

3.2.1 State Space. The compound vector input is widely used in RL-based ship collision avoidance research. The state is represented with the information of the own-ship and the target-ships in a vector form. To keep the input vectors unified in size, only the nearest target-ship would typically be included, even in the multiple target-ship encounters situation. In this study, since we are discussing the potential features in the waypoint approaching and collision avoidance work process, the whole working space, including the information of the own-ship and all the target-ships, needs to be taken into consideration. To further explore the work feature transferability, we also tried to indicate more work process feature-related information, such as the risk assessment and applicable rules in the input state, to pass to the neural network. Thus, to construct the input state containing such information with unified size, the informatic game window generated by the gameplay simulation environment is treated as the input state, graphically describing the current state and providing visual information effectively. CNNs are leveraged as a feature extractor due to their power of spatial information capturing.

3.2.2 Action Space. After passing the input state to the neural network, the learning agent, i.e., the own-ship, takes actions based on the output prediction to approach the waypoint while avoiding collisions with encountering target-ships. Each action is assigned a command heading angle change of -25 to 25 deg from the agent's current angle. The action space design is shown in Table 1.

During the training, the agent will choose one of the seven actions listed in Table 1 to adjust the heading angle accordingly. In typical RL tasks, an action predicted by the neural network can be completed and evaluated by the reward function within one step. Thus, the neural network and environment can interact with every step. In this study, however, the chosen action cannot be completed and evaluated in a single step due to the ship dynamics. The large moment of inertia of the own-ship causes a significant delay between the time of taking the action of a command angle and the time the ship's heading reaches that angle. The delay can span several action steps. Before converging to the command angle, the environment is updated continuously based on changing angular velocity and position. Still, the updated windows cannot be sent to the neural network to make the next action prediction since the current action has not been done, resulting in the delay of model–environment interaction. To solve this delay issue, when the agent starts to choose an action a, the initial state s, which is the gameplay simulation window at that moment, is recorded. At each step, the completion of the chosen action is judged. When the remaining angle difference is less than a certain threshold, 1 degree in this study, the action is considered to be completed. The state s' is generated by the game environment at that moment is recorded. In the middle process between starting and finishing, the action choice is accessible to the environment to update the ship's position and velocity. Still, it is omitted for the neural network training. The reward function is adjusted accordingly to handle the delay in action completion. The reward is cumulated step by step for each action choice from the starting point to the completion. The summation result is treated as the reward r_a for the chosen action a. The experience $\{s, a, r_a, s'\}$ is sent to the experiment buffer to train the neural network later. The state s is updated with s' for the next-step action prediction. The graphical explanation is shown in Fig. 2.

3.3 Feature Extraction and Finetuning. The transfer RL method is applied in this study to explore work process feature transferability. Feature transfer is conducted between one base case and several target cases. Despite initialization variations, the base case and target cases follow the same RL pipeline, as shown in Fig. 3, which allows the neural network to interact with the environment and be trained by perceiving feedback. The neural network in the base and target cases is called the base and target networks.

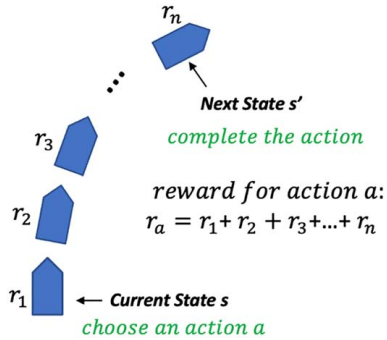

Fig. 2 Reward design handling action completion delay

They are all constructed by four convolution layers, working as a feature extractor, and two fully connected layers, mainly focusing on decision-making. The environment constructs training scenarios based on the settings of the base case and target cases, passing the generated game window to the network as the input state. The network makes a prediction and forwards it to the environment to update the work process accordingly. Once the action is completed, the reward feedback will be sent back to the network, and the game window will be updated for training and initializing the next iteration.

The base network is trained with the baseline model in the base case. The checkpoint is saved with trained weights, which will be reused in the target cases training. Feature extraction and finetuning methods have been used for the target case training. The target network is initialized with parameters transferred from the base network, which have been pretrained in the base case. The transferred parameters are frozen as a fixed feature extractor for the feature extraction method while open to be updated in the training process for the finetuning method. Besides the whole network transfer, different layer-level transfer RL is also implemented to explore the transferability of work process features through the layer level of the neural network. Figure 3 uses two convolution layers' transfer, for example, describing the initialization and training process of layer-level transfer RL. "2" refers to the former two convolution layers closer to the input to allow general feature transfer. The first two layers are initialized with parameters transferred from the base network to keep frozen or not according to the applied method. Other layers are uniformly initialized. Other layer-level transfer follows the same process; only the number of transferred layers changes accordingly. After initialization, the target network interacts with the environment and gets updated during training. The target case training is also conducted with the baseline for comparisons, which is uniformly initialized and trained from scratch in target cases and does not include any transferred parameters. Thus, the transferability of the work process feature can be explored comprehensively through different combinations between the frozen setting and layer level.

4 Case Study

The case studies selected for this research derive from the Imazu problems [43], which are well-regarded in the field of ship handling and encapsulate a wide range of realistic ship encounter scenarios. In typical ship navigation systems, there are both global and local planners. Our system functions as the local planner, tasked with short-range waypoint navigation and collision avoidance, while waypoints are designated by the global planner involved in the broader navigation strategy. We analyze four pairs of cases; each pair consists of a shared base case and a distinct target case. This setup allows for the application of the transfer RL approach, facilitating the extraction and subsequent reuse of work process features from the base case training in various target cases, which helps illuminate the transferability of these features.

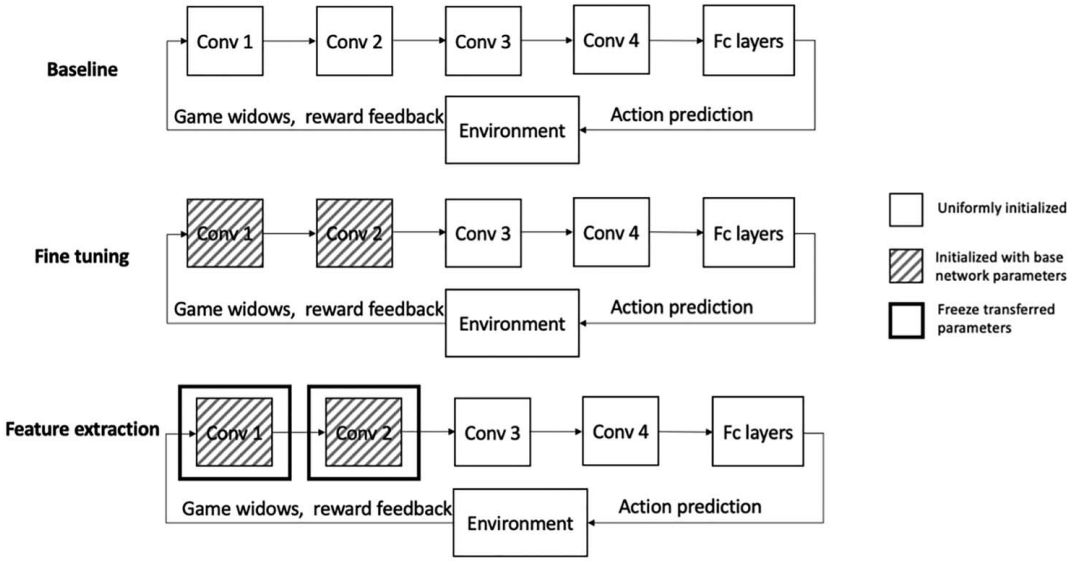

Fig. 3 Baseline and two transfer RL methods

For the implementation details, we developed a gameplay-like environment for ship navigation using Pygame. This environment includes a hardcoded simulation of ship dynamics and incorporates specific vehicle traffic rules to enhance realism. The neural network, essential for processing and learning from this simulation, is constructed using TensorFlow 2. The first convolutional layer convolves 32 8×8 filters with stride 4. The second convolutional layer convolves 64 4×4 filters with stride 2. The third convolutional layer convolves 64 3×3 filters with stride 1. The fourth convolutional layer convolves 200 7×7 filters with stride 1. The following fully connected layers are designed with a dueling structure.

Section 4.1 outlines the specific work process features and settings for each case. Section 4.2 details the parameters of the ship domain, risk assessment practices, and pertinent traffic regulations. Based on these elements, Sec. 4.3 elaborates on the design of the reward function.

4.1 Work Process Features. To study the features in this specific work process, we first define the work process features from different aspects. Graphic features through moving image frames in work processes can be treated as one kind of feature. On the other hand, some nongraphic information highly related to the work process can also be potential features. In this study, we conclude ship dynamics, number of target-ships, heuristic knowledge of risk assessment, DCPA, and applicable COLREG, which come from crucial components of the ship collision avoidance process, to be nongraphic features of the work process. Other than that, there may be other hidden features formed by certain combinations or some other way, which cannot be concluded but do exist and distinguish the work process. To make the environment fully informative using controllable computation resources, we use simple geometries to cover the crucial features in the ship collision avoidance process, facilitating the feature transfer purpose. All these nongraphical features are represented by geometrics in the gameplay image frames and extracted by the neural network, along with graphic features during training. As a result, all the features are forwarded and transferred between layers of the neural network through screen frame input.

To explore the effect of variation of features and similarity on work process features transferability, the study cases vary transferred features and similarity when constructing base case and target case pairs. The common base case is designed with particular dynamics and a single applicable COLREG rule. For each case pair, the target case is constructed by varying one or more features from the base case while keeping the rest the same. Complexity increases, and similarity decreases when changing features. Cases 1 and 2 change a single feature from the base case, ship dynamic feature, and rule-related feature. They share the same similarity with the base case, which will be used to investigate the effect of different features. From cases 3 to 4, each target case adds more complexity based on case 2, while similarity with the base case decreases, which will be used to investigate the effect of similarity. The base case and all target cases are illustrated by the start state, borrowed from Imazu problems, and the path plotting, constructed by the predefined paths of target-ships and the after-training path of own-ship as shown in Fig. 4.

The own-ship dynamics are changed in case 1; the DCPA and applicable COLREG remain the same as the base case since the target-ship setting is unchanged. Meanwhile, a smaller ship with less moment of inertia and rudder gain compared to the ship dynamics described in Sec. 2 is selected. Specifically, the moment of inertia is 50; the rudder gain is 0.05. Since the own-ship becomes lighter, it has less delay and becomes capable of achieving the desired angle in fewer steps. Thus, more action choices are required in the same waypoint approach process. In this way, by reusing the neural networks (NNs) pretrained in the base case with the larger ship in the target case with the smaller ship, the effect of the ship dynamic difference can be reflected in the transfer RL process.

For case 2, the ship dynamics and the target-ship 1 are set to be the same as the base case. Meanwhile, an extra target-ship moving toward the own-ship from the front is introduced. We call it target-ship 2. Target-ship 2 shares the same velocity as target-ship 1. Based on the starting position and velocity, target-ship 2 does not change the DCPA of the own-ship. When target-ship 2 enters the collision avoidance zone of the own-ship, to avoid collision with target-ship 2, the own-ship should follow "head-on" COLREG rules by taking actions of turning right. It is similar to how the own-ship takes actions to avoid collision with target-ship 1. Generally, target-ship 2 also needs to turn right in this situation, but in this case study, only the own-ship is trained to learn collision avoidance and rules following strategies; the target-ships just dumbly move forward with predefined direction and speed.

For case 3, the ship dynamics and target-ship 1 are the same as the base case; target-ship 2 comes from the left-hand side of the own-ship. It changes the DCPA of the own-ship because it is closer to the own-ship than target-ship 1. According to COLREG rules, the own-ship has road right; it can keep "stand-on," and target-ship 2 should turn right to "give away." However, target-ship

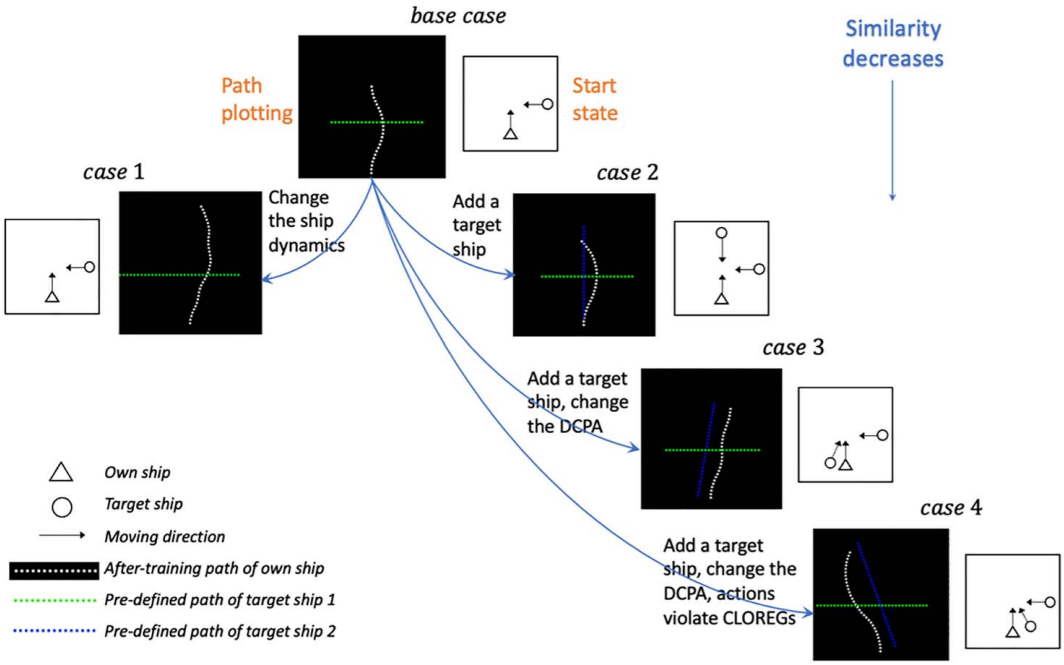

Fig. 4 Base case and target cases

2 always follows the predefined direction and speed, and it is already in the mandatory collision avoidance zone of the own-ship. The own-ship should make turns to avoid collisions instead of standing on. For target-ship 1, the own-ship should turn right when it enters the collision avoidance zone to give away the road right because it comes from the right side of the own-ship. Considering both target-ships, the learning agent should learn to avoid collisions by turning right.

Case 4 describes the rule violation situation. The ship dynamics and target-ship 1 are the same as the base case; target-ship 2 comes from the right side of the own-ship, also in the mandatory collision avoidance zone of the own-ship. DCPA of the own-ship is changed due to target-ship 2. Based on the COLREG rules, the own-ship should turn right when encountering target-ship 1 and target-ship 2 because these two target-ships both come from its right side. But target-ship 2 is too close to the own-ship; if the own-ship turns right, a collision will happen between the own-ship and target-ship 2. So, the own-ship will learn to take mandatory collision avoidance actions against the COLREG to avoid collisions in the training process. Although some penalties will be applied for violating COLREG rules with target-ship 1 in the mid-process, the largest penalty of collision can be avoided, and the largest reward for reaching the waypoint will be gained at the last step. The learning agent should figure out the logic of temporary sacrifices and a big final reward to nail this case.

4.2 Feature Representation. To minimize the risk of collisions, we designed the own-ship domains to facilitate risk assessment during work processes based on literature [20,44,45]. Accordingly, the simulation RL environment is constructed with the relative coordinates of the own-ship, as shown in Fig. 5, which is also close to the real ship navigation system. The white line represents the current heading path of the own-ship to the waypoint, located at the end of the path. The green polygon represents the encountering target-ship. There are encounters of multiple ships in study cases, but only one target-ship is used in the figure for illustration. The calculation of other target-ships follows the same procedure. During the training, the own-ship was fixed on the origin in the waypoint reaching and collision avoidance process. The goal and target-ships moved relatively to the own-ship. At each step, the position and rotation of the ships are updated based on real-time data. The coordinates of the goal and target-ships were converted from the absolute coordinates using rotation and transition matrixes. Nongraphic features like collision risks, COLREG, and ship dynamics are represented geometrically by the domains and models described below.

4.2.1 Minimal Ship Domain. To access potential collisions, the concept of a minimal ship domain is applied in this research [44]. The collision is judged if the target-ship or other obstacles enter this domain. Ship minimal domain is defined as an ellipse with the major axis along the own-ship's moving direction and the minor axis perpendicular to it, as the blue polygon shown in Fig. 1. As detailed in the literal [44], the half-length of the major and minor axes is set to be 4L and 1.6L, with L the ship length. Ships typically have a longer stopping distance when moving

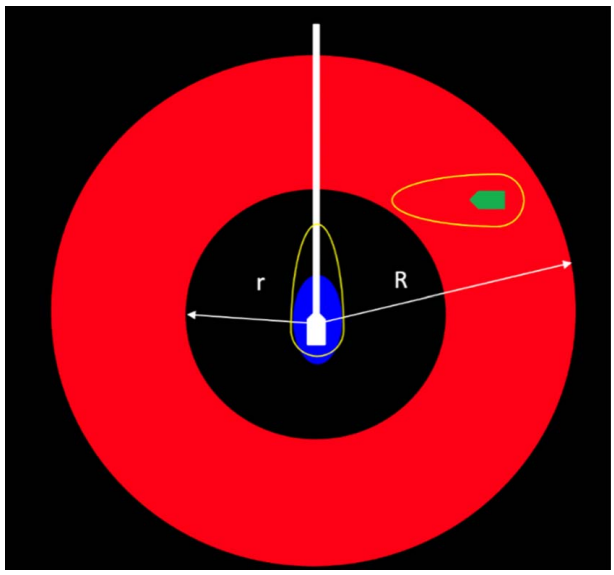

Fig. 5 Gameplay window

forward and backward compared to moving sideways. Therefore, a longer major axis provides a larger safety buffer in the direction of travel, allowing for adequate reaction time to avoid collisions. Additionally, a larger ship has a greater moment of inertia, requiring more distance to stop than smaller ships. Thus, the ship domain is related to the length of the ship. In this study, the own-ship length is set to be 300 m, and then, the half-length of the major axis and minor axis of the ship domain is 1200 m and 480 m, respectively. The distance between the own-ship and the target-ship in this area also satisfies the collision avoidance passing distance of 0.3 nm determined according to ship maneuvering [46] (Fig. 6).

4.2.2 COLREG Domain. While reaching a waypoint and avoiding collisions, the learning agent, i.e., the own-ship, needs to follow the COLREG rules. An event trigger mechanism is introduced to further define the stage of applying COLREG rules [20]. The large red circle radius R represents the boundary value of the collision risk stage (5 nm) suggested by the guide to the Rules for maritime collision avoidance. The small black circle radius r represents the minimal close-quarters situation 2 nm [47]. When the distance between the own-ship and target-ship target_dist is larger than R, the target-ship is in the safe zone of the own-ship. The own-ship can take any direction to reach its waypoint. When r<target_dist<R, the target-ship enters the collision avoidance zone, and the own-ship needs to take actions following COLREG rules to avoid collision with target-ship. When target_dist<r, the target-ship enters the mandatory collision avoidance zone, and the own-ship is allowed to take any emergent actions to avoid the collision, even violating the COLREG rules.

4.2.3 Bumper Model. The bumper-shaped area outside the own-ship and target-ship indicates the area where the ship tends to keep clear of static or moving obstacles. The shorter axis and longer axis are 1.6 times and 6.4 times the length of the own-ship, respectively. The bumper model was designed to trigger the collision avoidance action [45]. In this study, since we use another method to distinguish the emergency of collision and action modes, the bumper area is depicted in the environment to provide more information in the input image. For example, the rotation of the target-ship bumper shows the ship dynamics and the overlapping of the own-ship bumper and target-ship bumper indicates the risk of collision. Although the target-ship's relative movement of the own-ship is shown in the input image, the ship size is relatively small compared to the water area, so it might be hard for the NNs to observe. Based on our testing results, adding a bumper model could accelerate and stabilize the training. The NNs became capable of capturing more knowledge through more informatic image input.

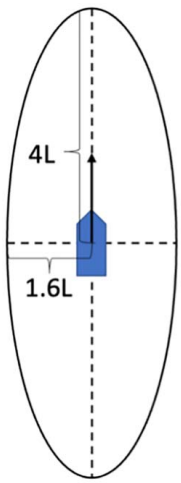

Fig. 6 Minimal ship domain

4.3 Compound Reward Function. As mentioned in Sec. 3.4, the reward of each chosen action is cumulated step by step from the initial state to the final state. This section will give more details about the reward function design for every single step. A compound reward function is used in this study. Besides the final positive or negative reward, subtle rewards are added based on the training purpose to enrich the sparse reward space [27]. For each step of a chosen action, compound reward constructed by episodic reward, shaping reward, and constant reward is applied to evaluate the status of this certain step, as shown in Eq. (12). Comprehensive knowledge from different sources is incorporated to guide the agent training. The learning goal and restriction are implied by the episodic reward and constant reward. Meanwhile, more reward signals are provided by shaping reward components to enrich the sparse reward space, encouraging the training to converge to the desired direction more efficiently:

$$R_{\text{tot}} = R_{\text{goal}} + R_{\text{col}} + R_{\text{colregs}} + R_{\text{dist}} + R_{\text{dev}} \tag{12}$$

In the above equation, R_{tot} is the compound reward for each step, constructed with five components. $R_{\text{goal}} = 1000$ and $R_{\text{col}} = -1000$ will be given if the agent reaches the waypoint or collides with target-ships; the episode will end at the same time. Otherwise, a 0 value will be assigned. The value of this episodic reward is determined by experimental results after comparing different parameter settings.

R_{colregs} guides the agent to follow the COLREG rules during the collision avoidance process. For each target-ship, the value varies due to different situations related to own-ship COLREG domain, as shown in the following. If multiple target-ships appear, R_{colregs} of each target-ship will be summed up.

When target_dist>R:

$$R_{\text{colregs}} = 0 \tag{13}$$

When r<target_dist<R:

$$\begin{cases} \text{if the own ship follows COLREGs: } R_{\text{colregs}} = -5 \\ \text{if the own ship voilates COLREGs: } R_{\text{colregs}} = 2 \end{cases} \tag{14}$$

When target_dist<r:

$$R_{\text{colregs}} = 0 \tag{15}$$

R_{colregs} guides the agent to follow the COLREG rules during the collision avoidance process. For each target-ship, the value varies due to different situations related to the own-ship's COLREG domain, as shown in the following. If multiple target-ships appear, R_{colregs} of each target-ship will be summed up.

If the target-ship is in the safe zone of the own-ship, there is no need to take collision avoidance actions yet, $R_{\text{colregs}} = 0$; when the target-ship is detected to enter the own-ship collision avoidance zone, the collision avoidance mechanism of the own-ship is triggered, and the applicable COLREG rules will be determined by the functions embedded in the environment. At each step, if the own-ship owns the road right, the "stand-on" scenario, for example, the maneuver of own-ship will not be restricted. If the own-ship does not own the road right, like encountering crossing-give-away or head-on situations described in Sec. 2, the own-ship will be trained to give away the road right to encountering target-ships with the R_{colregs} term. To be detailed, if the agent turns right, following the COLREG rules, a small reward will be given, $R_{\text{colregs}} = 2$. If the action violates the COLGREs rules, like turning left or keeping the current direction, a -5 penalty will be assigned with R_{colregs}. The value of reward and penalty are also decided based on several trials; this combination works for different cases. With this setting, the agent is not only able to learn to avoid collision following the rules but also capable of dealing with emergent situations even against the rules. If, for some reason, the target-ship enters the mandatory collision avoidance zone of the own-ship, there is not much space left for the agent to adjust the direction.

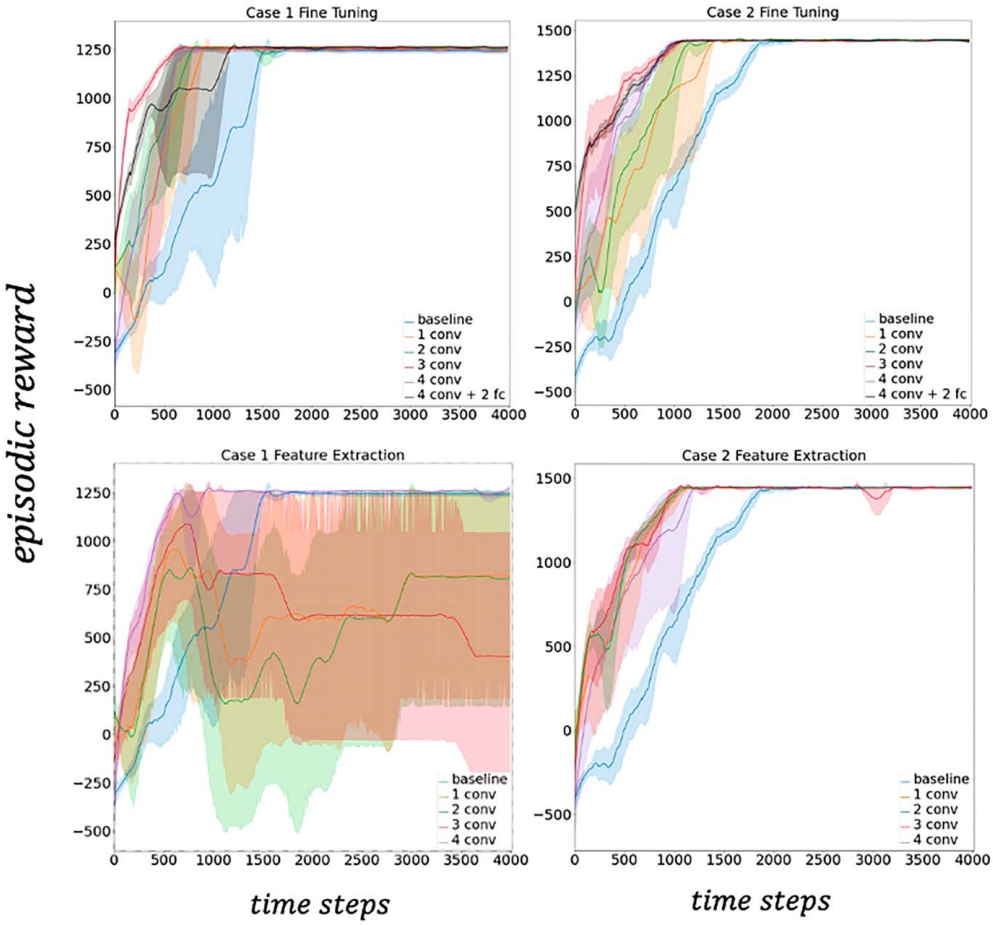

Fig. 7 The reward of case 1 and case 2 using the different layer-level transfer learning strategies

Driving safety is more important than rules following; the own-ship can take any action to avoid collisions, even violating COLREG rules. R_{colregs} is set to 0 to remove rule restrictions.

The shaping reward components R_{dist} and R_{dev} are set to be relatively small in order to provide subtle guidance at each step. R_{dist} is related to the distance between the own-ship and the desired waypoint, forcing the agent to move closer to the waypoint. If the distance is shortened at the current step compared to the last step, R_{dist} is positive. Otherwise, R_{dist} is negative. R_{dev} is defined based on the angle between the heading direction of the own-ship and the direction of the straight line connecting the own-ship and the goal. If the angle is 0, the agent moves directly toward the waypoint, R_{dev} is maximum. In the following equations, k_1 and k_2 are set to 0.5 and 2.5, respectively. This setting makes the shaping components not dominate over episodic and COLREG-related components, which should be major ones, nor vanish due to the small value. Distance rewards and deviation rewards are also balanced during the training process:

$$R_{\text{dist}} = k_1 \times (\text{prev\_dist} - \text{curr\_dist}) \tag{16}$$

$$R_{\text{dev}} = k_2 \times \exp\left(-\text{abs}(\text{dev})\right) \tag{17}$$

The reward shaping is based on heuristic knowledge of how we expect the learning agent to behave during training. Linear shaping is used by R_{dist}. Exponential shaping is used by R_{dev}. Linear shaping continuously encourages the agent to reduce the distance to the waypoint during the whole process. The reward changes linearly with the moving-forward distance, regardless of how far the agent is from the waypoint. Exponential shaping is introduced to R_{dev} to add more gradient near the peak to distinguish the peak and the neighboring high reward points, leading the agent to achieve the best solution (deviation is 0 deg) instead of stopping at relatively good solutions. Also, based on our experimental results, exponential shaping is verified to perform better than linear shaping or other shaping like sinusoidal for R_{dev}.

5 Results and Discussion

The neural network was pretrained in the base case and reused in target cases by applying two transfer RL methods: feature extraction and finetuning. The target cases' training results differ due to the transfer of different work process features. Different layer levels of transfer RL are also explored. The transferability of work process features through the layer level of the neural network is shown by comparing the transfer learning results between four target cases. The reward function of training cases during the training process is shown in Figs. 7 and 8. The x-axis represents the number of training episodes; the y-axis represents the cumulated reward for each episode. Each layer level of transfer RL is plotted by 10 random seeds, which are used to initialize the exploration rate generator, allowing the agent to encounter diverse trajectories. The solid line represents the mean of all samples; the shading area records the variation. The legends represent different layer levels. For example, "1 conv" in finetuning results represents transferring the first layer of the pretrained neural network to the target network and finetuning the network during the target case training process. In feature extraction cases, it refers to freezing the first layer of the target network after carrying out parameter transfer and then updating the rest. Especially, "4 conv + 2 fc" in finetuning results refers to transferring the parameter of all convolution layers and fully connected layers of the pretrained network to the target

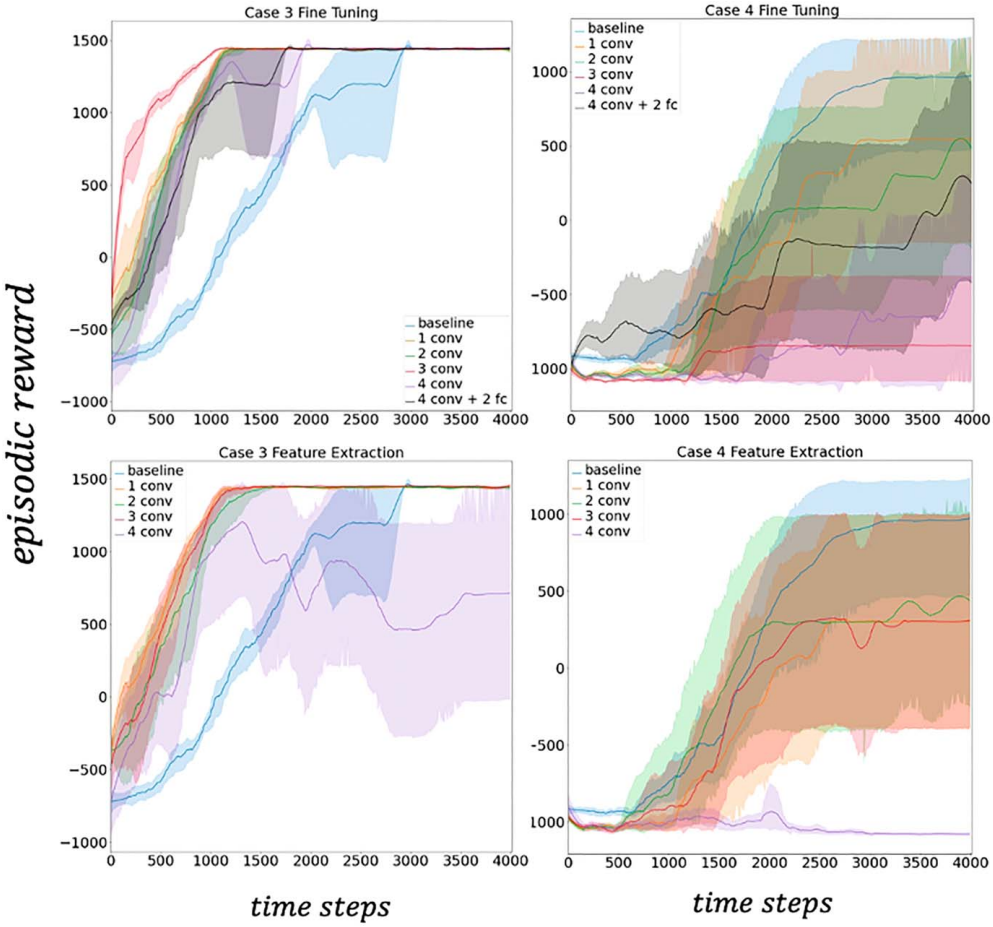

Fig. 8 Reward of case 3 and case 4 using the different layer-level transfer learning strategies

network and then finetuning all of them during the training process. It is meaningless to freeze all the parameters during training, so this setting is omitted for the feature extraction method. Also, for each target case, the transfer learning results are compared with the baseline, which is trained from scratch without being initialized by pretrained parameters.

5.1 Effect of Feature Categories. The training results of the finetuning method for two target cases, case 1 and case 2, are very similar. While regarding the feature extraction method, transfer RL leads to very different results. Since both cases have minor changes compared with the base case, the similarity between the two cases and the base case is approximately equivalent. The transfer learning performance should be close from a similar point of view. The variation of different features may be the main reason for different training results. For case 1, the ship dynamic feature is changed. Features transfer is conducted between the base case with a larger ship moment of inertial and rudder gain and the target case with a lighter ship. For case 2, an extra target-ship is added to the target case to vary the rule-related feature. Features are extracted from a single-rule case to a double-rule case.

5.1.1 Effect of Feature Categories Using Finetuning. For case 1 and case 2, the similarity with the base case is high. Although dealing with different kinds of features, variation in feature category almost does not affect the feature transferability. The finetuning method enhances the training efficiency compared to the baseline. Higher layer-level transfer converges faster than lower layer-level transfer, except for a slight performance drop for whole network transfer in case 1.

For the pretrained base network, lower layers extract general features of the ship driving process; higher layers extract features specific to the base case, which may be far from the target case if the two cases are very unlike. Since within both case pairs, the similarity is high. The base case shares most features with the target case. More pretrained layer transfer leads to higher training efficiency. As shown in the figure, the three convolution layers extract more features shared by the base case and the target cases than the one and two convolution layers, leading to the fastest convergence. Four convolution layers' transfer works not as well as three convolution layers' transfer due to some base case-specific features. But the variation of feature category does change the final action choice of the learning agent, thus affecting the performance of the whole network transfer.

Regarding the shape of the own-ship after-training path in Fig. 4, base case and case 2 look very similar, while case 1 looks different. The change in ship dynamics in case 1 causes more effect on decision-making than the newly introduced COLREG rule in case 2. Due to this, the whole network transfer, including the fully connected layers (related to decision-making), leads to lower reward and slower convergence in case 1 than in case 2; longer training is needed to fill the gap between different decision-making strategies.

5.1.2 Effect of Feature Categories Using Feature Extraction. For the feature extraction method, the reward function plotting of case 1 and case 2 looks different. As mentioned before, the variation in work process features is a key factor causing the large difference.

In case 2, all convolution layer-level transfer works better than the baseline since the similarity between the base case and case 2 is high and the decision-making strategy is close; the general and

specific features extracted in the base case by the transferred layers both fit the target case's work process. For one and two convolution layers' transfer, the feature extraction method works better than the finetuning method; since the general features are almost the same, there is no need to process finetuning. All layer-level transfers do not have obvious differences; only four convolution layers' transfer shows slightly lower rewards than other lines due to the specificity caused by higher layer transfer.

Different from case 2, four convolution layers' transfer work best in case 1 with the feature extraction method; the accumulated reward is equivalent to the reward obtained by the finetuning method in Fig. 7. Other layers' transfer level shows higher reward than the baseline at the beginning, while a large continuous variation happens instead of reaching the highest reward step by step. It infers that the dynamic features cannot be re-learned if the transferred convolution layers are frozen. The frozen part is separated from the rest of the model, which will be updated during training. The overall finetuning for the convolution part of the neural network is needed to finetune the different ship dynamic features transferred from the base case. Alternatively, it is also possible to transfer the entire four layers of the neural network and freeze them while only finetuning the decision-making process related to the fully connected layers. This approach also works when the ship dynamic difference between the base and target cases is not too high, as shown in case 1. Even though the convolution layers carry different dynamic features, the decision-making layers are finetuned to cover the differences.

In summary, when similarity is high for rule-related features, a higher level of layer transfer is preferable for the feature extraction method to transfer more shared features and save computational resources. When dealing with different ship dynamic features, all layer levels do benefit the learning process at the very beginning. Still, the ship dynamic features cannot eventually be transferred through separate convolution layers even if the ship dynamic difference is not high. Another thing that needs to be mentioned is we only explored a small change in ship dynamics within the same simulating system; the conclusion may change if a large difference is involved or a different dynamic system is introduced. For example, transferring the four convolution layers with the feature extraction method may not work well.

5.2 Effect of Case Pair Similarity. Based on case 2, the rule-related features are modified in cases 3 and 4 to decrease the similarity with the base case. The training result of these two cases shows an obvious difference, although the ship dynamics are the same. The effect of different similarities between the case pairs is reflected. For case 3, the DCPA is changed by adding another target-ship, but the added ship has less road priority and does not affect the own-ship to take actions following COLREG rules. For case 4, the added target-ship changes the DCPA and forces the own-ship to take mandatory collision avoidance actions that violate COLREG. In this way, features extracted by the base network pretrained in the base case are effective in case 3, while they can be harmful in case 4 due to the large similarity changes.

5.2.1 Effect of Similarity Using Finetuning. By applying the finetuning method to case 3, transfer RL enhances the training efficiency compared with the baseline. The three-layer transfer performs best for all levels, extracting more features shared by base and target case. The reward starts to drop with four convolution layers and the whole model transfer. The higher layer-level finetuning does not work as well as the lower level because the higher level of the pretrained network can extract features more specific to the base case than the target cases. Compared to case 2, the similarity is decreasing, the higher layer transfers fewer shared features in case 3, and more training time is required to finetune the target network to deal with case-specific features. Also, the overall trend of converging lines for four convolution layers and the whole model transfer looks similar to the baseline, which is trained from scratch, indicating that the decision-making strategies transferred by the higher layer do not help target case training due to lower similarity. A similar finetuning process to learning from scratch is needed to develop the decision-making strategies for this target case.

For case 4, the similarity with the base case further decreases, and significantly different action choices appear; the transferred features do not fit this work process and even harm the training. Even though the first and second convolution layers extract the general features from the base case, they cannot properly initialize the network and nail the target case. Thus, the finetuning method does not work better than the baseline anymore. As the number of transferred convolutions increases, the reward decreases accordingly. The fourth layer transfer and whole model transfer work less optimally in case 4; the reason may relate to features specific to the base case and decision-making. The three convolution layer transfer works best in other cases and works worst in case 4. It may indicate the third layer processes rule-related features, which can be crucial to the work process. Based on the experimental results, the poor initialization of the third convolution layer leads to unpromising training performance.

5.2.2 Effect of Similarity Using Feature Extraction. When applying the feature extraction method in cases 3 and 4, convolution layer transfer shows significant performance drops compared to other lines due to the specificity of the transferred features. Even though we verified that the three convolution layer transfer extracts the most shared feature between the base case and target case, it thus shows the highest performance with the finetuning method. When applying the feature extraction method, keeping the first three convolution layers frozen through the training process does not work better than the one and two convolution layers' transfer anymore. That may also be caused by the case-specific features. Only four convolution layers are involved in this study; the third layer is relatively close to the output, which can be considered a relatively high-level layer in this case. Although adding the third convolution layer to the transfer learning makes it possible to extract most shared features, it may also extract some features specific to the base case, which does not show advantages without finetuning.

For case 4, because the similarity between the base cases is too low, the transferred features show side effects even with finetuning, and the same thing happened for the feature extraction method. Compared to case 1, 1–3 convolution layer's transfer has similar effects with case 4; the performances are nearly equivalent. In comparison, the four convolution layer transfer shows the lowest reward finally, which has the higher and highest performance in case 1 and case 2 when similarity is high, further proving the influence of case–pair similarity on deeper convolution layer transfer.

6 Conclusions and Future Work

This article explores the transferability of work process features in the context of ship collision avoidance. We enhanced the realism and applicability of our RL simulations by incorporating detailed ship dynamics and adjusting the model–environment interaction to account for the inertia-related delays. Defined work process features were employed to generate and analyze case studies, investigating their impact and the influence of case-pair similarity on feature transferability. Our evaluation involved two transfer RL methods: feature extraction and finetuning, benchmarked against a baseline. Key conclusions drawn from our experiments include the following.

- Feature Transferability: Transferability using the finetuning method appears invariant to feature categories unless significantly different decision-making strategies are required. However, with the feature extraction method, ship dynamic features proved nontransferable through isolated frozen convolution layers when any dynamic modification occurred.

- Rule-Related Features: In our neural network configuration, rule-related features seemed to be predominantly transferred by the third convolution layer. This layer performed well in cases adhering to COLREG rules but was less effective in scenarios where rules were violated.
- Layer Specificity: Deeper layer transfers, which extract features more specific to the base case, are preferable when there is a high similarity between base and target cases, particularly if their decision-making processes are aligned. Conversely, their effectiveness decreases with reduced similarity.
- Impact of Similarity: The similarity between case pairs notably influences the transferability in deeper convolution layers, regardless of the type of features processed.

The demonstrated transferability of work process features underscores the potential of transfer RL to address complex engineering challenges by strategically reusing critical features. However, this study's scope was limited to comparing ship dynamic and rule-related features, which may restrict its broader applicability. Our future research will extend this approach to a wider range of engineering tasks and features to enhance the versatility and effectiveness of transfer learning strategies.

Acknowledgment

This article is based on the work supported by the Autonomous Ship Consortium (ASC) with members of BEMAC Corporation, ClassNK, MTI Co., Ltd., Nihon Shipyard Co. (NSY), Tokyo KEIKI Inc., and National Maritime Research Institute of Japan. Chuanhui Hu, a Ph.D. student in our lab, provided the ship dynamic model. The authors are grateful for their support and collaboration on this research.

Conflict of Interest

There are no conflicts of interest. This article does not include research in which human participants were involved. Informed consent is not applicable. This article does not include any research in which animal participants were involved.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N., 2021, "Stable-Baselines3: Reliable RL Implementations," J. Mach. Learn. Res., **22**(1), pp. 12348–12355.
[2] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M., 2013, "Playing Atari With Deep RL," arXiv:1312.5602.
[3] Todorov, E., Erez, T., and Tassa, Y., 2012, "Mujoco: A Physics Engine for Model-Based Control," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Algarve, Portugal, Oct. 7–12.
[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., 2017, "Proximal Policy Optimization Algorithms," arXiv:1707.06347.
[5] Suh, N. P., 2007, "Ergonomics, Axiomatic Design and Complexity Theory," Theor. Issues Ergon. Sci., **8**(2), pp. 101–121.
[6] Zhu, Y., Tang, T., Zhao, S., Joralmon, D., Poit, Z., Ahire, B., Keshav, S., et al., 2022, "Recent Advancements and Applications in 3D Printing of Functional Optics," Addit. Manuf., **52**, p. 102682.
[7] Weiss, K., Khoshgoftaar, T. M., and Wang, D., 2016, "A Survey of Transfer Learning," J. Big Data, **3**(1), pp. 1–40.
[8] Bengio, Y., 2011, "Deep Learning of Representations for Unsupervised and Transfer Learning," Proceedings of ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, July 2.
[9] Deniz, E., Şengür, A., Kadiroğlu, Z., Guo, Y., Bajaj, V., and Budak, Ü., 2018, "Transfer Learning Based Histopathologic Image Classification for Breast Cancer Detection," Health Inf. Sci. Syst., **6**(1), pp. 1–7.
[10] Nixon, M., and Aguado, A., 2019, *Feature Extraction and Image Processing for Computer Vision*, Academic Press, Cambridge, MA.
[11] Liu, X., and Jin, Y., 2020, "RL-Based Collision Avoidance: Impact of Reward Function and Knowledge Transfer," AI EDAM, **34**(2), pp. 207–222.
[12] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., 2014, "How Transferable Are Features in Deep Neural Networks?" Adv. Neural Inf. Process. Syst., **27**, pp. 3320–3328.
[13] Wang, X., and Jin, Y., 2022, "Work Process Transfer RL: Feature Extraction and Finetuning in Ship Collision Avoidance," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, St. Louis, MO, Aug. 14–17, Vol. 86212, American Society of Mechanical Engineers.
[14] Ahmed, O., Träuble, F., Goyal, A., Neitz, A., Bengio, Y., Schölkopf, B., Wüthrich, M., and Bauer, S., 2020, "Causalworld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning," arXiv:2010.04296.
[15] Zhao, W., Queralta, J. P., and Westerlund, T., 2020, "Sim-to-Real Transfer in Deep RL for Robotics: A Survey," 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, Dec. 1–4 .
[16] Goerlandt, F., and Kujala, P., 2014, "On the Reliability and Validity of Ship–Ship Collision Risk Analysis in Light of Different Perspectives on Risk," Saf. Sci., **62**, pp. 348–365.
[17] Hu, C., and Jin, Y., 2023, "Long-Range Risk-Aware Path Planning for Autonomous Ships in Complex and Dynamic Environments," ASME J. Comput. Inf. Sci. Eng., **23**(4), p. 041007.
[18] Li, G., Yang, Y., Zhang, T., Qu, X., Cao, D., Cheng, B., and Li, K., 2021, "Risk Assessment Based Collision Avoidance Decision-Making for Autonomous Vehicles in Multi-Scenarios," Transp. Res. C: Emerg. Technol., **122**, p. 102820.
[19] Fan, Y., Sun, Z., and Wang, G., 2022, "A Novel RL Collision Avoidance Algorithm for USVs Based on Maneuvering Characteristics and COLREGs," Sensors, **22**(6), p. 2099.
[20] Li, L., Wu, D., Huang, Y., and Yuan, Z. M., 2021, "A Path Planning Strategy Unified With a COLREGS Collision Avoidance Function Based on Deep RL and Artificial Potential Field," Appl. Ocean Res., **113**, p. 102759.
[21] Chuah, J., 2009, *Law of International Trade: Cross Border Commercial Transactions*, Sweet & Maxwell Ltd., London, UK.
[22] Wang, X., Yadav, V., and Balakrishnan, S. N., 2007, "Cooperative UAV Formation Flying With Obstacle/Collision Avoidance," IEEE Trans. Control Syst. Technol., **15**(4), pp. 672–679.
[23] Park, J.-W., Oh, H.-D., and Tahk, M.-J., 2008, "UAV Collision Avoidance Based on Geometric Approach," 2008 SICE Annual Conference, The University Electro-Communications, Japan, Aug. 20–22, IEEE.
[24] Xia, J., Zhu, X., Liu, Z., Luo, Y., Wu, Z., and Wu, Q., 2022, "Research on Collision Avoidance Algorithm of Unmanned Surface Vehicle Based on Deep Reinforcement Learning," IEEE Sens. J., **23**(11), pp. 11262–11273.
[25] Meyer, E., Heiberg, A., Rasheed, A., and San, O., 2020, "COLREG-Compliant Collision Avoidance for Unmanned Surface Vehicle Using Deep Reinforcement Learning," IEEE Access, **8**, pp. 165344–165364.
[26] Sawada, R., Sato, K., and Majima, T., 2021, "Automatic Ship Collision Avoidance Using Deep Reinforcement Learning With LSTM in Continuous Action Spaces," J. Mar. Sci. Technol., **26**(2), pp. 509–524.
[27] Huang, B., and Jin, Y., 2022, "Reward Shaping in Multiagent RL for Self-Organizing Systems in Assembly Tasks," Adv. Eng. Inform., **54**, p. 101800.
[28] Nomoto, K., 1960, "Analysis of Kempf's Standard Maneuver Test and Proposed Steering Quality Indices," 1st Symposium on Ship Maneuverability, Annapolis, MD, May.
[29] Zhao, L., and Roh, M.-I., 2019, "COLREGs-Compliant Multiship Collision Avoidance Based on Deep RL," Ocean Eng., **191**, p. 106436.
[30] Sivic, J., and Zisserman, A., 2003, "Video Google: A Text Retrieval Approach to Object Matching in Videos," IEEE International Conference on Computer Vision, Nice, France, Oct. 13–16.
[31] Harris, A., and Jones, S.H., 2016, *Writing for Performance*, Sense Publishers, Rotterdam, The Netherlands., pp. 19–35.
[32] Liu, K., Cheng, Y.-Q., and Yang, J.-Y., 1993, "Algebraic Feature Extraction for Image Recognition Based on an Optimal Discriminant Criterion," Pattern Recognit., **26**(6), pp. 903–911.
[33] Yin, Y., Tran, M., Chang, D., Wang, X., and Soleymani, M., 2023, "Multi-Modal Facial Action Unit Detection With Large Pre-Trained Models for the 5th Competition on Affective Behavior Analysis in-the-Wild," CVPR 2023: 5th Workshop and Competition on Affective Behavior Analysis in-the-wild (ABAW), Vancouver, British Columbia, Canada., June 19.
[34] Wagner, J., Schiller, D., Seiderer, A., and André, E., 2018, "Deep Learning in Paralinguistic Recognition Tasks: Are Hand-Crafted Features Still Relevant?" Interspeech 2018, Hyderabad, India, Sept. 2–6.
[35] Lowe, D. G., 1999, "Object Recognition From Local Scale-Invariant Features," Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, Sept. 20–27, Vol. 2, IEEE.
[36] He, K., Zhang, X., Ren, S., and Sun, J., 2016, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
[37] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., 2015, "Going Deeper With Convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
[38] Chen, Z., Zou, H., Wang, Y., Liang, B., and Liao, Y., 2017, "A Vision-Based Robotic Grasping System Using Deep Learning for Garbage

Sorting," 2017 36th Chinese Control Conference (CCC), Dalian, China, pp. 11223–11226.
[39] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., et al., 2022, "Rt-1: Robotics Transformer for Real-World Control at Scale," arXiv:2212.06817.
[40] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S., 2014, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Columbus, OH, June 24–27.
[41] Jin, Y., 2019, "ISC: Intelligent Situation Awareness and Collision Avoidance," Technical Report.
[42] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S., 2018, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep RL With a Stochastic Actor," International Conference on Machine Learning, Stockholm, Sweden, July 10–15.
[43] Imazu, H., 1987, "Research on Collision Avoidance Maneuver," Ph.D. thesis, The University of Tokyo (in Japanese), Tokyo.
[44] Goerlandt, F., Montewka, J., Lammi, H., and Kujala, P., 2011, "Analysis of Near Collisions in the Gulf of Finland," *Advances in Safety, Reliability and Risk Management*, F. Goerlandt, J. Montewka, H. Lammi, and P. Kujala, eds., CRC Press, Boca Raton, FL, pp. 2880–2886.
[45] Shen, H., Hashimoto, H., Matsuda, A., Taniguchi, Y., Terada, D., and Guo, C., 2019, "Automatic Collision Avoidance of Multiple Ships Based on Deep Q-Learning," Appl. Ocean Res., **86**, pp. 268–288.
[46] International Maritime Organization, 1972, *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*, International Maritime Organization, London, UK.
[47] Fujii, Y., 1980, "A Definition of the Evasive Domain," Navigation, **65**, pp. 17–22.