

Large Language Model Guided Reinforcement Learning for Maritime Ship Collision Avoidance

Yasmin Pang¹, Arsyi Aziz², Alex Zongo¹, Peng Wei¹, and Kyle Crandall¹

¹Department of Mechanical and Aerospace Engineering, ²Department of Computer Science

George Washington University, Washington, D.C., USA

{Yasmin_pang, arsyia, a.zongo, pwei, crandallk}@gwu.edu

Abstract—Autonomous surface vessels require collision-avoidance systems that can adapt to multi-vessel encounters while remaining traceable to the International Regulations for Preventing Collisions at Sea (COLREGs). Conventional rule-based planners encode these regulations as predetermined maneuvers, providing structured and interpretable behavior that becomes increasingly conservative in crowded waters. Conversely, deep reinforcement learning may reduce conservatism, but yields uninterpretable policies whose safety and rule compliance are correspondingly difficult to verify. To bridge this gap, this paper presents a hybrid architecture within the CORALL framework to restore decision-level traceability. A large language model (LLM) processes an environmental state against encoded COLREGs regulatory guidelines to classify encounters and issue discrete navigational intents. A low-level reinforcement learning policy then issues heading and speed commands for a single ownship and is directionally constrained by COLREGs-based LLM intent guidance. Across a range of encounter geometries, this pipeline enabled successful path-following, collision-free ownship trajectories. The results indicate that pairing language-based reasoning with deep-learning-based planning can improve collision-avoidance efficiency while keeping maneuvers attributable to COLREGs.

I. INTRODUCTION

Safe and efficient collision-avoidance is critical for autonomous maritime navigation in multi-vessel encounters. Collision-free maritime navigation between two vessels is governed by the International Regulations for Preventing Collisions at Sea (COLREGs), which establish responsibilities for maneuvering in head-on, crossing, and overtaking scenarios [1]. Translating these regulations into autonomous control is challenging because the rules assign qualitative obligations, not specific numerical trajectories. The challenge becomes more pronounced in dense, multi-vessel encounters, where an action that resolves one conflict may create another.

Classical collision-avoidance methods include geometric-based planners and velocity-obstacle methods, which use metrics such as time to closest point of approach (TCPA) and distance to closest point of approach (DCPA) to evaluate the future possibility of a collision from a current trajectory [2]–[4]. Such classical analyses establish a foundation of methods used to assess potential collisions with basic physics, leveraging access to data that most ships’ traditional Automatic Identification Systems (AIS) broadcast, including the positions and speeds of other maritime vessels or obstacles within range.

Optimization-based approaches such as evolutionary algorithms [5], model predictive control [6], [7], and control

barrier functions [8] have been used to enhance collision-free navigation performance. These methods provide generally well-performing behavior, but can be difficult to define, and their operational computational cost and constraint complexity increase in multi-vessel encounters. In addition, COLREGs compliance is typically represented through geometric constraints, safety margins, or cost-function penalties rather than explicit rule classification. Thus, the basis for a maneuver’s compliance can be difficult to interpret, limiting transparent human-in-the-loop validation.

Reinforcement learning (RL) is a learning algorithm in which an agent learns optimal behavior that maximizes a reward function through repeated interactions in an environment. This optimal behavior is encoded in a learned policy that an agent follows. Unlike other optimization-based methods that re-plan trajectories at every time step, an RL policy can be trained beforehand in simulation. During execution, it queries the policy to predict the next optimal action, enabling rapid decision-making across varying encounter geometries. Previous studies have successfully applied Deep Reinforcement Learning (DRL) and Proximal Policy Optimization (PPO) to COLREGs-oriented path following and multi-ship collision-avoidance [9]–[14]. However, because the policies are black-boxes, their underlying representations resist formal verification and often lack interpretability.

To address interpretability limitations, Large Language Models (LLMs) offer interpretable reasoning aligned with rule-based frameworks and have recently been explored for more verifiable COLREGs-compliant decision-making [15]–[17]. While promising, LLM-based systems can behave unreliably and often require external tools to ensure safe operation. These tools, such as rule- and model-based planners, are useful as safety constraints, but can be difficult to scale in dense, dynamic, multi-vessel encounters [3], [11].

Thus, the integration of DRL and LLMs has been suggested as a promising research direction to leverage the strengths of each approach within the dynamic needs of unmanned maritime vehicle operation [9]. This work presents an LLM-guided RL architecture for maritime collision-avoidance by integrating a learned RL-based local planner into the CORALL (COLREGs-Guided Risk-Aware LLM) framework [15]. The proposed approach replaces the rule-based reactive collision-avoidance local planner module with an RL policy whose proposed action is constrained by high-level navigation intent,

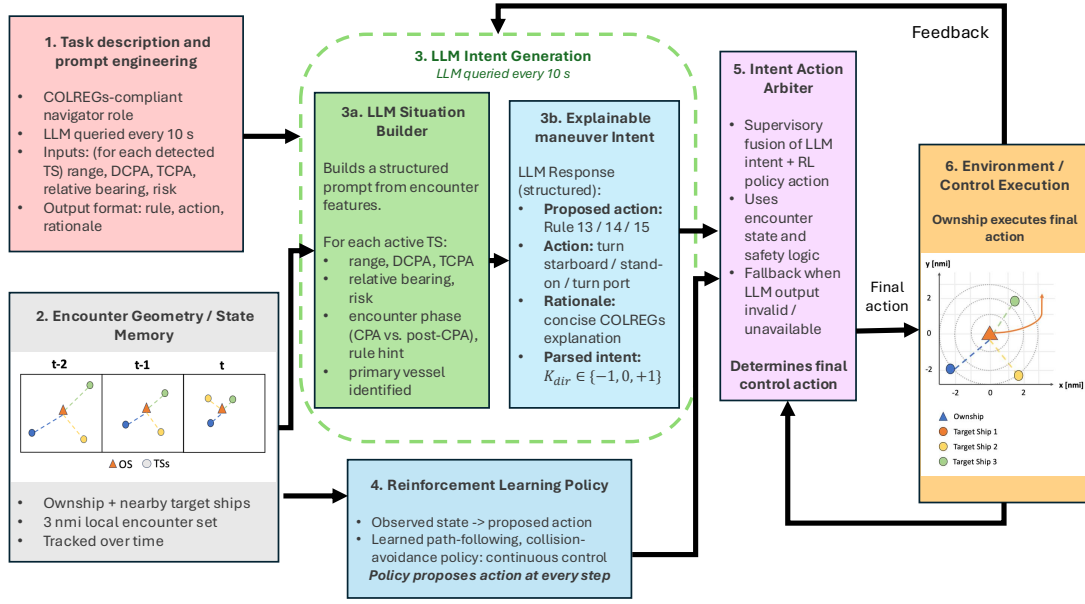


Fig. 1. Design-time architecture of the proposed LLM-guided reinforcement learning pipeline.

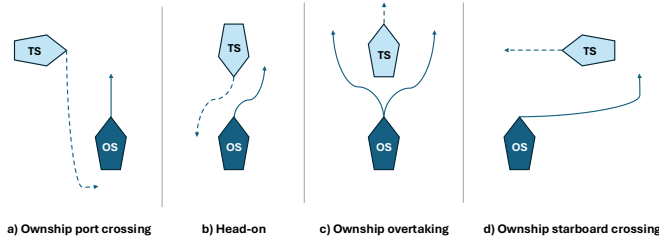


Fig. 2. COLREGs-defined basic two-vessel encounter maneuvers.

thus preserving COLREGs-based high-level decision-making. The proposed architecture improves operational efficiency while maintaining collision-free, safe separation behavior in multi-agent encounter scenarios.

II. METHODOLOGY

Our work builds upon the CORALL framework [15] and the proposed extension is visualized in Fig. 1. The primary distinction of our proposed framework lies in the execution layer: while CORALL relies on a rule-based local planner that computes collision risk using deterministic Z-shaped membership functions [15], our architecture replaces this module with an RL policy that continuously controls vessel heading and speed in a simulated maritime environment. Similar to [15], the proposed architecture employs an LLM to interpret the geometric-based Velocity Obstacle (VO) parameters of simulated maritime encounters and to infer maneuver intent (e.g., turning to starboard, giving way, or maintaining course), as illustrated in Fig. 2. The learned policy operates at the execution layer through a supervisory arbitration layer, whereby the high-level intent directionally constrains the RL policy's

proposed speed and turning control commands provided to the autonomous vessel to be COLREGs-compliant.

An LLM (GPT-4o) is queried through LangChain using the ChatOpenAI API at 10 s intervals to update high-level maneuver intent without requiring language-model inference at the same controller rate as the ownship. This sampling frequency was selected to avoid over-saturating the model with data and to mitigate control chatter. As visualized in Fig. 1, each query prompts the LLM to make a COLREGs-compliant decision as a ship navigator, based on VO parameters relative to target ships (range, DCPA, TCPA, relative bearing, and risk) [15] within a 5556 m (3 nmi) region around the ownship (OS). It returns a high-level maneuver decision, the corresponding COLREGs rule, and supporting rationale. This intent is parsed into a discrete directional variable, $K_{dir} \in \{-1, 0, +1\}$, representing a turn port, stand on, or turn starboard maneuver, respectively. The most recent valid intent is retained between LLM updates and passed, together with the RL policy's proposed action and current encounter state, to a supervisory arbitration layer. When validated LLM guidance is available, the arbiter constrains the heading component of the RL policy proposal to remain consistent with K_{dir} . If guidance is unavailable, stale, or rejected by the safety logic, the RL policy is retained, while a deterministic emergency supervisor retains highest control authority. Thus, the LLM provides supervisory maneuver intent.

The maritime simulation environment adopted from CORALL is governed by simplified first-order Nomoto vessel dynamics described in [15]. Within this environment, the RL policy controls an OS that interacts with dynamic target ships (TS), scripted to follow their initialized encounter trajectories. The simulations use a time step of 0.5 s and run for a maximum horizon of 900 s during experiments. A simulated

episode terminates if the OS collides with any obstacle TS. At the beginning of each episode, the OS and TSs are assigned predefined trajectories. All vessels were initialized with a nominal speed of 10 m/s along their prescribed trajectories. The TSs then maintain their trajectories while the OS follows its learned policy.

The RL policy is trained using PPO [18] across simulated vessel encounters involving one to three TSs. PPO was implemented using the Stable-Baselines3 "MlpPolicy", with separate actor and critic branches each comprising two fully connected hidden layers of 256 neurons with tanh activation. The actor maps the encounter observation state at each time step to discrete heading and speed actions, while the critic estimates the state value. To reflect information available from onboard navigation systems and AIS, the observation space combines vessel-state variables and relative motion quantities. Continuous position and velocity variables were normalized using predefined environment bounds, while angular variables were represented using their sine and cosine components, yielding bounded and comparably scaled inputs to the control policy. As described in Eq. (1), the observation at time t comprises the OS state $\mathbf{o}_t^{\text{OS}}$ defined in Eq. (2), the OS's relative geometry to its goal trajectory defined in Eq. (3), and its relative position and velocity with respect to each TS in the environment, defined in Eq. (4).

$$\mathbf{o}_t = [\mathbf{o}_t^{\text{OS}}, \mathbf{o}_t^{\text{goal}}, \mathbf{o}_t^{\text{TS}_1}, \mathbf{o}_t^{\text{TS}_2}, \mathbf{o}_t^{\text{TS}_3}], \quad (1)$$

$$\mathbf{o}_t^{\text{OS}} = [\tilde{x}, \tilde{y}, \sin \psi, \cos \psi, \tilde{r}, \tilde{u}_x, \tilde{u}_y, \tilde{b}], \quad (2)$$

$$\mathbf{o}_t^{\text{goal}} = [\sin \beta_g, \cos \beta_g, \tilde{d}_g], \quad (3)$$

$$\mathbf{o}_t^{\text{TS}_j} = [\tilde{\Delta x}_j, \tilde{\Delta y}_j, \sin \beta_j, \cos \beta_j, \tilde{\Delta u}_{x,j}, \tilde{\Delta u}_{y,j}]. \quad (4)$$

In Eqs. (2), (3), and (4), $(\tilde{\cdot})$ denotes a continuous quantity normalized using the corresponding environment bounds before being provided as input to the RL policy. The OS state comprises its position $[x, y]$, bearing $[\sin \psi, \cos \psi]$, velocity $[u_x, u_y]$, yaw rate r , and a bias term, b , representing a steering/yaw disturbance in its environment [15]. Also, β_g describes the OS's relative bearing to the final waypoint, and d_g its distance from that waypoint. Similarly, β_j denotes the relative bearing from the OS to another TS, while Δx_j , Δy_j , $\Delta u_{x,j}$, and $\Delta u_{y,j}$ represent the corresponding relative position and velocity components.

Given this observation, the OS selects an action from the discrete action space:

$$\mathcal{A} = \mathcal{A}_\psi \times \mathcal{A}_u, \quad |\mathcal{A}_\psi| = 7, \quad |\mathcal{A}_u| = 5, \quad (5)$$

which consists of a heading-offset command and a forward surge speed command. The heading offset, $\Delta\psi$, is selected from seven discrete values spanning $[-25^\circ, 25^\circ]$ relative to the nominal waypoint-following heading. The surge-speed is selected from five discrete values:

$$u \in \{7.0, 8.25, 9.5, 10.75, 12.0\} \text{ m/s}. \quad (6)$$

The reward function balances path-following completion with collision-avoidance and safe vessel separation, and is

defined in Eq. (7). At each time step t in the environment, the reward for agent (i) is defined as:

$$r_{i,t} = \begin{cases} -r_{\text{col}}, & d_{i,t}^{\min} < L \\ r_{i,t}^{\text{progress}} - r_{i,t}^{\text{safe}} + r_{i,t}^{\text{goal}}, & \text{otherwise.} \end{cases} \quad (7)$$

where L is the vessel length overall (LOA), $d_{i,t}^{\min}$ is the minimum instantaneous separation between agent (i) and any other vessel, and $r_{\text{col}} = 10,000$ is the terminal collision penalty.

The route-progress component of the reward is defined in Eq. (8). Here, $w_p = 200$ is the progress-reward coefficient, and $\rho_{i,t}$ denotes the agent's normalized progress toward the final waypoint. Thus, positive values of $\Delta\rho_{i,t}$ reflect forward progress along the route, as conveyed in Eq. (9). The scaling factor $\alpha_{i,t}$ reduces this incentive during unsafe maneuvers.

$$r_{i,t}^{\text{progress}} = \alpha_{i,t} w_p \Delta\rho_{i,t}, \quad (8)$$

$$\Delta\rho_{i,t} = \rho_{i,t} - \rho_{i,t-1}, \quad \rho_{i,t} = \text{clip}\left(\frac{p_{i,t}}{\ell_i}, 0, 1\right). \quad (9)$$

In Eq. (9), $p_{i,t}$ is the projected progress traveled along the route from its origin, and ℓ_i is the total route length. To discourage aggressive route completion during hazardous encounters, the progress reward is scaled according to Eq. (10). An active conflict, denoted by $\mathcal{C}_{i,t}$ in Eq. (11), is identified when another vessel is either within $5L$ of the agent or is predicted to pass within $4L$ during the next 120 s. In such cases, the progress reward is reduced by 50%. Here, $d_{i,t}^{\text{CPA}}$ and $\tau_{i,t}^{\text{CPA}}$ denote the minimum predicted DCPA and TCPA respectively.

$$\alpha_{i,t} = \begin{cases} 0.5, & \mathcal{C}_{i,t} \\ 1.0, & \text{otherwise,} \end{cases} \quad (10)$$

$$\mathcal{C}_{i,t} = (d_{i,t}^{\min} < 5L) \vee (0 < \tau_{i,t}^{\text{CPA}} < 120 \wedge d_{i,t}^{\text{CPA}} < 4L). \quad (11)$$

The safety component of the reward function provides a bounded penalty, as shown in Eq. (12), that increases continuously as vessel separation decreases or a predicted CPA becomes hazardous.

$$r_{i,t}^{\text{safe}} = \min\left(C_{\max}, \frac{1}{2}q_{D,i,t}^2 + 2q_{C,i,t}^2 + q_{\text{CPA},i,t}\right), \quad (12)$$

where $C_{\max} = 3$ limits the total penalty. Such penalty combines two one-sided distance-based terms, $q_{D,i,t}$, $q_{C,i,t}$ with the normalized CPA-risk term $q_{\text{CPA},i,t}$.

The distance-based separation penalty terms $q_{D,i,t}$, $q_{C,i,t}$ are described in Eq. (13).

$$q_{D,i,t} = \left[\frac{5L - d_{i,t}^{\min}}{5L}\right]_+; \quad q_{C,i,t} = \left[\frac{2L - d_{i,t}^{\min}}{2L}\right]_+, \quad (13)$$

where $[x]_+ = \max(0, x)$.

Thus, $q_{D,i,t}$ introduces a gradual penalty when another vessel enters the $5L$ danger region, while the more heavily weighted $q_{C,i,t}$ term increases the penalty inside the $2L$ critical-separation region.

The predicted CPA contribution is given by Eq. (14), wherein $\mathbb{1}_{\mathcal{P}_{i,t}}$ is an indicator function that is equal to one when the CPA-based danger condition $\mathcal{P}_{i,t}$, defined in Eq. (15), is satisfied and zero otherwise. The term $q_{\text{CPA},i,t}$ increases as the predicted CPA distance decreases and as the predicted time to CPA becomes shorter.

$$q_{\text{CPA},i,t} = \mathbb{1}_{\mathcal{P}_{i,t}} \cdot \text{clip} \left(\frac{4L - d_{i,t}^{\text{CPA}}}{4L}, 0, 1 \right) \cdot \text{clip} \left(\frac{120 - \tau_{i,t}^{\text{CPA}}}{120}, 0, 1 \right), \quad (14)$$

$$\mathcal{P}_{i,t} = \{0 < \tau_{i,t}^{\text{CPA}} < 120 \wedge d_{i,t}^{\text{CPA}} < 4L\}. \quad (15)$$

Successful route completion produces a graduated terminal reward shown in Eq. (16) based on the minimum actual separation maintained between OS and TS over the episode. Here, $s_i^{\min}$ is the minimum vessel separation observed over the episode. The success bonus is applied only when the vessel reaches its final waypoint without collision. This graduated structure distinguishes between collision-free trajectories with different safety margins, providing a stronger incentive for successful trajectories that maintain greater separation from surrounding vessels.

$$r_{i,t}^{\text{goal}} = \begin{cases} 1000, & s_i^{\min} \geq 3L \\ 700, & 2L \leq s_i^{\min} < 3L \\ 300, & s_i^{\min} < 2L \\ 0, & \text{goal not reached.} \end{cases} \quad (16)$$

During training, encounters were created by randomizing the relative bearings the OS would encounter different TSs, specifying a desired encounter point. This ensures a predefined encounter occurs, enabling the agent to learn appropriate behavior in such scenarios. The OS is assigned dense trajectories spanning 3704 m (2 nmi), which are used for both training and evaluation. To start, the OS mainly observes two-vessel encounters to learn the fundamental collision-avoidance behavior in the most basic, COLREGs-prescribed scenarios. As training progressed, a curriculum was implemented so that the OS would experience higher traffic densities and learn more difficult and safer maneuvers.

Following training, pre-defined encounter scenarios were used to evaluate the learned policy performance. These scenarios reproduced the encounter geometries previously used by CORALL to evaluate a rule-based planner, but with closer waypoints [15]. The corresponding qualitative trajectories are shown in Fig. 3.

To evaluate the performance of the RL-based planner against the rule-based planner, for each test case, the OS was initialized at a speed of 10 m/s and placed on a trajectory that

is consistent with the training environment, with dynamic, scripted TSs entering encounters with the OS at specified intersection points. The evaluation set included head-on encounters, overtaking situations, and a variety of crossing encounters from the starboard and port sides of the OS perspective. Key metrics to compare planner performance include the OS total path length, total travel time, minimum separation distance experienced with another TS, and risk exposure $Risk_{\text{exp}}$. Collision risk was evaluated using the composite risk formulation adopted from CORALL [15], which combines normalized DCPA, TCPA, and instantaneous vessel range using Z-shaped membership functions [19]. For each OS-TS pair, the resulting score $Risk_{ij}(t) \in [0, 1]$ represents the instantaneous collision risk at time t . To capture the duration and severity of hazardous encounters over an episode, this work extends the instantaneous CORALL metric through a cumulative risk-exposure defined in Eq. (17).

$$Risk_{\text{exp}} = \sum_k \max_{j \neq i} Risk_{ij}(t_k) \Delta t. \quad (17)$$

At each time step, the maximum pairwise risk represents the most critical encounter involving OS agent i . Unlike the maximum instantaneous risk, which captures only the peak severity of an encounter, this cumulative measure accounts for the duration of sustained risk exposure. Lower values therefore correspond to trajectories that spend less time in high-risk configurations.

Finally, the reliability of the LLM supervisory layer was evaluated over 100 episodes for each of three representative Imazu encounter cases (Cases 1, 6, 18) from [15]. At each LLM decision interval, the encounter state, parsed maneuver direction $K_{\text{dir}} \in \{-1, 0, +1\}$, and applied controller action were logged. The parsed intent was then compared with a rule- and geometry-based maneuver derived from the corresponding encounter classification and phase. To avoid over-counting multi-vessel prompts, each LLM query was scored once using the highest-risk target associated with that decision interval. Across the three cases, 9,507 LLM calls were evaluated.

The valid intent output rate represents the fraction of LLM calls that returned a usable, parsed response that could be converted into a maneuver direction. Strict accuracy measures consistency with the deterministic COLREGs reference used to anchor the LLM prompt and was computed only for valid calls with a high-confidence, single-label reference action. A missed maneuver denotes a failure to issue a required avoidance action, whereas an unnecessary maneuver denotes an avoidance command issued when the reference action was to maintain the current course. Collision and success rates were evaluated separately over complete episodes, allowing LLM decision reliability to be compared with the resulting closed-loop navigation performance. The code for this work is available in the project repository.¹

RL vs CORALL Baseline Ownship Trajectories Across All 22 Imazu Cases

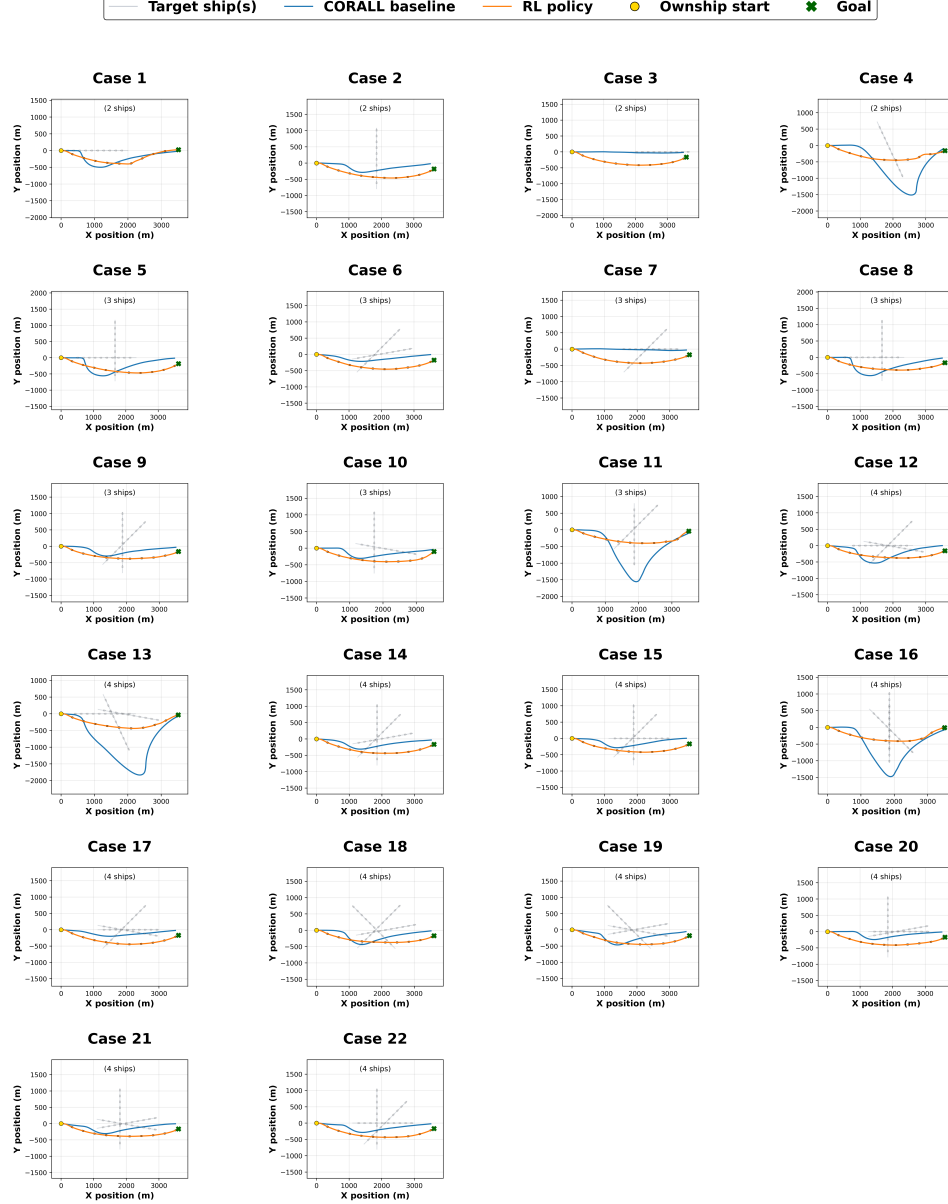


Fig. 3. Qualitative trajectories of RL-based and rule-based planned ownships against scripted target ships over encounter scenarios during evaluation.

III. RESULTS

To evaluate the efficacy of the proposed framework, the performance of the RL-based planner was compared against that of the rule-based planner using the scripted encounter cases presented in Fig. 3. The results of this comparison are shown in Fig. 4, which reports the average total path length and time traveled, minimum separation distance between the OS and any TS, and the mean cumulative risk exposure over 100 episodes, for each Imazu case. The results are averaged for encounters between two to four ships. Compared with the rule-based planner, the RL-based planner reduced route-completion

time by up to 54% and cumulative risk exposure by up to 37%, while maintaining relatively consistent minimum separation as encounter complexity increased. The corresponding qualitative trajectories of the RL-based OS and rule-based OS are shown in Fig. 3, for two-, three-, and four-ship encounters. Although the two planners produced slightly different paths, the RL-based OS generally used smoother and less abrupt turns while still executing effective collision avoidance maneuvers.

A second evaluation examined the reliability of the LLM integration and the resulting closed-loop performance. As summarized in Table I, 72.7% of all 9,507 LLM calls returned valid, parsable guidance, with substantial variation among the three cases (64.9–79.5%). Among high-confidence calls with

¹https://github.com/yaspang/RL_CORALL_extension

TABLE I
LLM INTEGRATION RELIABILITY AND CLOSED-LOOP PERFORMANCE

Case	Valid Intent	Strict Acc.	Missed	Unnecessary	Collision	Success
1	64.9%	100.0%	0.0%	0.0%	0.0%	100.0%
6	79.5%	91.9%	0.0%	8.1%	0.0%	100.0%
18	73.6%	96.6%	2.5%	0.9%	0.0%	100.0%
Overall	72.7%	96.0%	0.8%	3.1%	0.0%	100.0%

Note: $n = 9,507$ total LLM calls across three encounter cases. Strict accuracy and error rates are computed over the high-confidence single-label subset. Collision and success are policy-level metrics evaluated over 100 episodes per case.

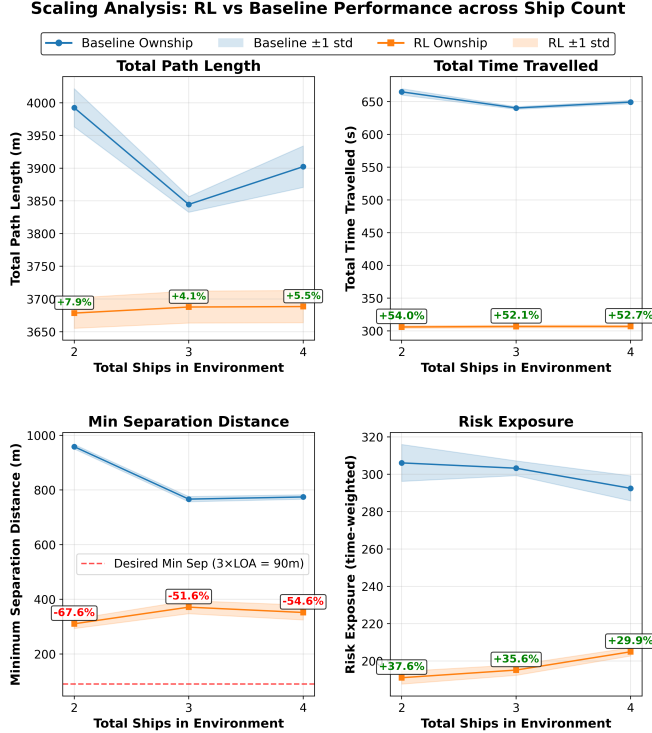


Fig. 4. Performance comparison between rule-based and reinforcement learning-based planner

a single reference maneuver, the LLM achieved 96.0% strict agreement, with missed and unnecessary maneuver rates of 0.8% and 3.1%, respectively. Despite occasional invalid outputs and mismatched guidance, the integrated controller successfully completed all 300 full pipeline evaluation episodes without collision.

IV. DISCUSSION

The comparative performance evaluation results in Fig. 4 suggest that replacing CORALL's deterministic reactive planner with a learned PPO policy can improve navigation efficiency, while maintaining collision-free safe separation from TSs. Qualitatively, the learned policy produced observably smoother and less abrupt avoidance trajectories than the rule-based baseline. These results indicate that the RL policy can exploit the heading and speed action space to resolve encounters with less conservative maneuvering, while the

reward formulation retains incentives for collision-avoidance, safe separation, and route progress.

The LLM evaluation demonstrates the feasibility of using interpretable COLREG-based intent to supervise learned, low-level maneuver execution. However, only 72.7% of calls yielded usable parsed outputs, reflecting an important limitation associated with the LLM output robustness, rather than control failure. The layered architecture prevented these failures from becoming a single point of failure by retaining the PPO policy and an emergency supervisor as fallbacks. Some intent-execution discrepancies may also reflect the multi-rate architecture, since the LLM was updated every 10 s, while the RL controller responded to encounter changes every 0.5 s; encounter event-triggered LLM updates may reduce this effect. Thus, collision-free, successful closed-loop operation across all evaluated episodes suggests that the layered architecture can tolerate imperfect LLM guidance by retaining the RL policy and safety supervisor as fallback mechanisms.

V. CONCLUSION

This work demonstrates the feasibility of integrating RL into a hierarchical LLM-guided framework for maritime collision-avoidance. The newly proposed architecture preserves decision-level traceability to COLREGs-based maneuver intent, while improving trajectory efficiency and reducing ownship risk exposure. The findings also highlight important trade-offs between safety, efficiency, and scalability in multi-vessel environments. Future work should focus on improving LLM reliability, while testing the full pipeline on a broader range of encounter geometries. Extending the framework toward multi-agent control represents a promising direction for scalable deployment. Overall, this approach provides a pathway towards interpretable and learning-enabled autonomy in safety-critical maritime systems.

VI. ACKNOWLEDGMENT

OpenAI ChatGPT (GPT-5) was used as an interactive research aid to support iterative refinement of the reinforcement-learning reward formulation described in this work. GitHub Copilot was used to assist with the development of data-visualization scripts and code supporting the LLM-integration pipeline. All AI-assisted suggestions and generated code were reviewed, modified, tested, and validated by the authors. The authors retain responsibility for the methodology, implementation, experimental analysis, and conclusions presented in this work.

REFERENCES

- [1] International Maritime Organization, “Convention on the International Regulations for Preventing Collisions at Sea (COLREGs),” <https://www.imo.org/en/about/conventions/pages/colreg.aspx>, 1972, accessed: 2026-04-21.
- [2] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using the relative velocity paradigm,” in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 1993, pp. 560–565 vol.1.
- [3] W. Zhang, C. Yan, H. Lyu, P. Wang, Z. Xue, Z. Li, and B. Xiao, “COLREGS-based path planning for ships at sea using velocity obstacles,” *IEEE Access*, vol. 9, pp. 32 613–32 626, 2021.
- [4] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, “Safe maritime navigation with colregs using velocity obstacles,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4728–4734.
- [5] C. Tam and R. Bucknall, “Path-planning algorithm for ships in close-range encounters,” *Journal of Marine Science and Technology*, vol. 15, no. 4, pp. 395–407, 2010.
- [6] P. Potočník, “Model Predictive Control for Autonomous Ship Navigation with COLREG Compliance and Chart-Based Path Planning,” *Journal of Marine Science and Engineering*, vol. 13, no. 7, p. 1246, Jun. 2025.
- [7] B.-O. H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, “Hybrid Collision Avoidance for ASVs Compliant With COLREGs Rules 8 and 13–17,” *Frontiers in Robotics and AI*, vol. 7, p. 11, Feb. 2020.
- [8] C. Lee, J. Park, and J. Kim, “Efficient COLREGs-Compliant Collision Avoidance Using Turning Circle-Based Control Barrier Function,” *Mechatronics*, vol. 117, p. 103495, Jul. 2026.
- [9] Y. Li, D. Wu, Z. You, G. Chen, and D. Wu, “Deep reinforcement learning for collision avoidance in unmanned surface vehicles: State-of-the-art,” *Applied Ocean Research*, vol. 164, p. 104778, 2025.
- [10] Y. Wang and Y. Zhao, “Multiple ships cooperative navigation and collision avoidance using multi-agent reinforcement learning with communication,” *Ocean Engineering*, vol. 320, p. 120244, 2025.
- [11] L. Zhao and M. I. Roh, “COLREGs-compliant multiship collision avoidance based on deep reinforcement learning,” *Ocean Engineering*, vol. 191, p. 106436, 2019.
- [12] P. Sarhadi, W. Naeem, and N. Athanasopoulos, “A survey of recent machine learning solutions for ship collision avoidance and mission planning,” *IFAC-PapersOnLine*, vol. 55, no. 31, pp. 257–268, 2022.
- [13] M. Parooei, M. T. Masouleh, and A. Kalhor, “Semi-Supervised Deep Reinforcement Learning in Decentralized Multi-Agent Collision Avoidance and Path Planning in a Complex Environment,” in *2023 31st International Conference on Electrical Engineering (ICEE)*, Tehran, Iran, May 2023, pp. 749–755.
- [14] D. Xue, D. Wu, A. S. Yamashita, and Z. Li, “Proximal Policy Optimization with Reciprocal Velocity Obstacle-Based Collision Avoidance Path Planning for Multi-Unmanned Surface Vehicles,” *Ocean Engineering*, vol. 273, p. 114005, 2023.
- [15] K. Agyei, P. Sarhadi, and W. Naeem, “CORALL: A COLREGs-Guided Risk-Aware LLM for Decision-Making in Maritime Autonomous Surface Ships,” *IEEE Journal of Oceanic Engineering*, pp. 1–19, 2026.
- [16] K. A. Christensen, A. Gusev, A. G. Tufte, O. A. Alsos, and M. Steinert, “AI captain: Conversational mission planning and execution system for autonomous surface vehicles,” *Ocean Engineering*, vol. 338, p. 121988, Nov. 2025.
- [17] H. Namgung, J.-S. Kim, and D.-U. Jang, “Path Planning and Collision Avoidance Technologies for Maritime Autonomous Surface Ships: A Review of COLREGs Compliance, Algorithmic Trends and the Navigation-GPT Framework,” *The Journal of Navigation*, vol. 78, no. 4–6, pp. 430–450, 2026.
- [18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [19] P. Sarhadi, W. Naeem, and N. Athanasopoulos, “An Integrated Risk Assessment and Collision Avoidance Methodology for an Autonomous Catamaran with Fuzzy Weighting Functions,” in *2022 UKACC 13th International Conference on Control (CONTROL)*. IEEE, 2022, pp. 228–234.