

Article

Bridging Symmetric Dynamics and Asymmetric Semantic Objectives: Runtime-Assured Predictive Safety Control for Autonomous Surface Vehicles

Manlin Wang ¹, Hongjun Tian ^{1,*} , Maoyuan Sun ², Yuhan Zhou ³, Shuai Huang ² , Jingwen Zeng ², Yang Xiong ¹, Yichen Li ¹, Yichen Wang ¹ , Yijie Yin ¹, Xiaoyin Guo ⁴, Jiani Wu ⁵, Jiesan Zhang ⁵ and Ying Tang ⁵

¹ College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China; 2523205@st.shou.edu.cn (M.W.)

² College of Energy and Mechanical Engineering, Shanghai University of Electric Power, Shanghai 201306, China

³ Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China

⁴ College of Information Technology, Shanghai Ocean University, Shanghai 201306, China

⁵ College of Oceanography and Ecological Science, Shanghai Ocean University, Shanghai 201306, China

* Correspondence: tianhongjun86@163.com

Abstract

In maritime navigation, vessel dynamics and open-water environments often exhibit inherent symmetries, whereas control objectives, particularly collision avoidance and COLREGs compliance, are strictly asymmetric, specifying unique responsibilities (e.g., give-way versus stand-on) and distinct desired trajectories. This paper proposes a runtime-assured, dual-envelope predictive safety-control framework for autonomous surface vehicles (ASVs) that directly addresses the symmetry and asymmetry in complex encounters. To manage asymmetric semantic objectives, a large language model (LLM) serves as a semantic-governance module, generating structured COLREGs labels (encounter type, vessel responsibility, and maneuver tendency). These semantic outputs are strictly validated before entering the control stack. In parallel, to break the dangerous symmetry of collision risks, vessel-motion prediction and uncertainty inflation construct a physical safety envelope. A deterministic MPC-CBF safety filter then computes admissible control commands, balancing the symmetric homogeneous tracking dynamics with asymmetric collision-avoidance constraints. A runtime assurance monitor supervises semantic validity and solver latency, preventing unsafe decisions. Simulation results demonstrate that the proposed intelligent decision-making and control scheme significantly improves rule-aware collision avoidance while reducing excessive conservatism, providing a profound perspective for viewing maritime safety through symmetric and asymmetric control theory.

Keywords: symmetry and asymmetry; intelligent decision-making; decision and control; autonomous surface vehicles (ASVs); large language model (LLM); model predictive control (MPC); maritime collision avoidance



Academic Editor: Shaojie Zhang

Received: 2 June 2026

Revised: 22 June 2026

Accepted: 26 June 2026

Published: 1 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Rapid advances in maritime autonomy, including Maritime Autonomous Surface Ships (MASSs) and unmanned surface vehicles (USVs), have attracted increasing interest in ocean surveillance, marine conservation, port patrol, search-and-rescue operations, and intelligent transportation. In these applications, navigation is not merely a path-following problem

but also a safety-critical decision-making process in dynamic and uncertain maritime environments [1–3].

Collision avoidance remains a central challenge for the deployment of autonomous ships in complex maritime settings. The International Regulations for Preventing Collisions at Sea (COLREGs) provide general rules for common encounters, including head-on, crossing, and overtaking situations. However, these rules contain situation-dependent semantic requirements, such as timely action, appropriate course alteration, safe distance, and good seamanship, which are difficult to encode as fixed mathematical constraints [4–6].

From the profound perspective of control theory, maritime collision avoidance can be fundamentally viewed through the lens of symmetry and asymmetry. The physical environment and the nominal kinematics of ASVs often exhibit homogeneous symmetries. However, the control objectives dictated by COLREGs are almost always asymmetric, specifying unique behavioral responsibilities, such as the clear distinction between a “give-way” vessel and a “stand-on” vessel in a crossing situation. Consequently, an effective intelligent navigation system must leverage symmetric control theory for scalable, stable path-following, while employing asymmetric control strategies to address heterogeneous constraints, ambiguous semantic rules, and complex global collision-avoidance goals.

Conventional techniques, including rule-based planning, artificial potential fields (APFs), velocity obstacles (VOs), and model predictive control (MPC), can perform well in structured settings; however, they remain limited in multi-ship interactions, complex traffic scenarios, and semantically ambiguous encounters [7–9]. Learning-based algorithms, such as deep reinforcement learning (DRL), multi-agent reinforcement learning (MARL), and hybrid optimization-learning methods, provide greater adaptability to dynamic navigation and path-planning problems; for example, DRL-based and BIT* + TD3-based frameworks have been investigated for autonomous-vessel decision-making and energy-efficient USV path planning [10,11]. However, their decision-making processes are often opaque and difficult to validate, raising safety concerns in high-stakes maritime operations.

LLMs offer a promising tool for high-level decision support because they can interpret natural-language rules and generate human-readable reasoning. Recent studies on maritime-specific LLMs and LLM-based USV mission planning suggest that language models may support high-level maritime navigation reasoning and task-level decision support [12,13]. More specifically, recent work has explored LLM-assisted COLREGs decision-making and navigation support for autonomous vessels. Nevertheless, LLMs were not designed to serve as safety-critical controllers. Their outputs may be semantically plausible but physically ungrounded, delayed, inconsistent, or dynamically infeasible; consequently, LLM outputs should be validated and filtered before influencing vessel control [14].

Recent advances in lightweight visual detection, perception, and sensor-fusion technologies have enabled autonomous systems to obtain richer environmental awareness and detect obstacles in real time. Lightweight YOLO-based detectors have been explored in resource-constrained visual-recognition tasks outside the maritime domain [15], while radar, LiDAR, camera, and AIS-related measurements can support maritime target detection, dynamic obstacle tracking, and situation awareness for safe navigation and collision avoidance [16–20].

This paper proposes a runtime assurance (RTA)-oriented dual-envelope safety mechanism to address these collision-avoidance challenges. In the proposed framework, the LLM acts as a semantic reasoning module that translates structured encounter information into auditable COLREGs-conformant labels, including encounter type, vessel responsibility, relevant rules, recommended maneuvers, and prohibited actions. These outputs are validated and then mapped to a soft semantic compliance envelope, while a predictive risk envelope

and an MPC-CBF safety filter ensure physical collision avoidance. When semantic outputs are unreliable, the runtime assurance monitor triggers conservative or emergency fallback actions, thereby decoupling semantic reasoning from physical safety guarantees.

1. We introduce a COLREGs-oriented semantic control architecture for ASV collision avoidance that restricts the LLM to high-level rule interpretation while preserving physical safety through a deterministic controller.
2. We develop a safety-verification and correction mechanism that prevents unsafe or dynamically infeasible LLM-generated decisions from being executed directly.
3. We evaluate the proposed method in canonical two-vessel encounters and multi-vessel scenarios to assess collision avoidance, COLREGs compliance, and decision interpretability.

2. Related Work

2.1. COLREGs-Compliant Collision Avoidance for Maritime Autonomous Surface Ships

Collision avoidance for Maritime Autonomous Surface Ships (MASSs) and ASVs is a fundamental requirement for safe autonomous navigation. Unlike generic robotic obstacle avoidance, maritime collision avoidance must account for COLREGs, which specify encounter-dependent responsibilities in head-on, crossing, overtaking, give-way, and stand-on situations. However, COLREGs were originally written for human navigators rather than autonomous control systems. Terms such as early and substantial action, safe speed, proper lookout, and ordinary practice of seamen are, therefore, difficult to translate into deterministic mathematical constraints.

Recent surveys have shown that COLREGs-compliant navigation is not merely a geometric collision-prevention problem but a coupled problem involving situation awareness, encounter classification, risk assessment, decision-making, trajectory planning, and control execution. Burmeister and Constapel evaluated the use of artificial intelligence, machine learning, and data-driven solutions for autonomous collision avoidance at sea [21]. Zhang et al. surveyed collision-avoidance navigation systems for MASSs and reviewed regulatory, perception, planning, and navigation challenges [22]. Hu et al. reviewed COLREGs-compliant ASV navigation from classical methods to learning-based approaches [23]. Chang et al. further analyzed MASS collision-avoidance studies using the Conventional Collision Avoidance Process model and identified functional gaps between academic approaches and conventional bridge practice [24].

These studies indicate that COLREGs compliance cannot be guaranteed by a single local planner or a single collision-risk metric. An effective autonomous collision-avoidance system should integrate rule interpretation, risk prediction, maneuver feasibility, and verifiable safety bounds. This observation motivates architectures in which high-level COLREGs reasoning is decoupled from low-level control execution: semantic rule interpretation may guide the system, but deterministic and auditable modules must retain safety authority.

2.2. Rule-Based, Artificial-Potential-Field, Velocity-Obstacle, and ORCA Methods

Early COLREGs-compliant collision-avoidance methods were commonly based on rule logic, ship-domain models, artificial potential fields (APFs), velocity obstacles (VOs), reciprocal velocity obstacles, and search-based planners. These methods are attractive because they are interpretable, easy to analyze, and directly related to selected COLREGs encounter categories. Ship-domain methods define protected regions around vessels, APF methods generate attractive and repulsive navigation fields, and VO-based methods identify relative-velocity regions that can lead to collision. Rule-based planners then select course or speed changes according to encounter type and collision-risk metrics, such as DCPA and TCPA.

APF-based maritime planners have been adapted to include COLREGs-compliant repulsive fields and deterministic real-time path planning. Lyu and Yin proposed a real-time path-planning approach with COLREGs constraints based on modified artificial potential fields [25], while Li and Zheng developed a real-time USV collision-avoidance planner based on field theory [26]. VO-based methods provide a complementary geometric perspective. Kuwata et al. designed a maritime VO algorithm that accounts for COLREGs when guiding USVs in dynamic environments [27]. ORCA, originally developed for multi-agent robotics, provides sufficient local conditions for reciprocal collision avoidance without communication and is therefore a useful reference for reactive geometric avoidance [28].

These methods form the conceptual basis of the rule-based COLREGs and APF/VO/ORCA baselines evaluated in this study. However, they typically rely on hand-crafted encounter classifiers, risk thresholds, and priority rules. Although transparent, they can become fragile in heavy traffic, ambiguous responsibility assignments, and uncertain COLREGs situations in which multiple rules may apply. This limitation motivates predictive optimization strategies that reason over future trajectories rather than reacting only to the current collision geometry.

2.3. Model Predictive Control and Predictive Collision Avoidance

Model predictive control (MPC) has become a prominent approach for maritime collision avoidance because it can account for vessel dynamics, actuator limits, prediction horizons, and multi-objective cost functions. MPC-based algorithms evaluate candidate future trajectories or control sequences and select actions that balance path following, risk reduction, maneuver smoothness, and COLREGs compliance. MPC therefore provides the methodological basis for the MPC-only baseline used in this study.

Johansen et al. proposed a simulation-based control-behavior selection method that predictively evaluates collision hazards while maintaining COLREGs compliance [29]. The method generates a finite set of candidate control actions, propagates them forward in time, and evaluates them according to collision risk and rule adherence. Eriksen et al. introduced the branching-course MPC algorithm as a short-term maritime collision-avoidance mechanism that is robust to obstacle-estimation uncertainty and satisfies applicable COLREGs rules [30]. Hagen et al. later explored scenario-based MPC with multiple decision stages as a predictive strategy for COLREGs-compliant ship collision avoidance [31].

Although MPC-based solutions provide predictive feasibility and dynamic constraint handling, they usually depend on predefined rule encodings and manually specified semantic criteria. The optimizer can evaluate risk and feasibility, but it cannot by itself resolve the linguistic ambiguity of COLREGs or explain why a particular regulatory interpretation should dominate in a compound encounter. MPC is therefore well suited as a deterministic planning and implementation layer, but safety-critical COLREGs-compliant navigation also requires semantic reasoning and runtime safety supervision.

2.4. COLREGs-Aware Trajectory Optimization and Rule-Constrained MPC

A related line of research directly incorporates COLREGs into trajectory-optimization or MPC objectives. Instead of selecting from a restricted set of rule-based actions, these methods represent rule compliance as mode-dependent costs, constraints, or penalties within an optimization problem. This formulation enables smoother and more dynamically feasible trajectories while explicitly representing own-ship obligations in head-on, crossing, overtaking, give-way, and stand-on scenarios.

Tsolakis et al. proposed a COLREGs-aware optimal trajectory method for ASVs using model predictive contouring control [32]. Their approach integrates relevant COLREGs rules into motion planning and verifies the resulting behavior in simulation. Similar

optimization-based methods formulate collision-avoidance constraints, ship domains, and rule-dependent penalties to avoid trajectories that are technically safe but rule-inconsistent. These studies motivate the COLREGs-MPC-CBF baseline used here, in which the planner must not only avoid collision but also follow rule-aware maneuver preferences.

Nevertheless, COLREGs-based optimization alone cannot provide complete safety reasoning. The optimizer may become infeasible in dense traffic, depend on manually selected rule thresholds, or generate locally feasible trajectories with a limited safety margin. Formal safety bounds and runtime guarantees are therefore required when rule-constrained MPC is used in safety-critical closed-loop operation.

2.5. Control Barrier Functions and MPC-CBF Safety-Critical Control

Control barrier functions (CBFs) provide a formal mechanism for enforcing safety constraints in real time. The CBF-QP framework developed by Ames et al. integrates safety requirements with performance objectives by expressing safety as a forward invariance of an admissible set and embedding the resulting conditions in a quadratic program [33]. CBFs are therefore attractive for safety-critical control because they can act as real-time safety filters that minimally modify nominal control commands while enforcing safety constraints.

Combining MPC and CBFs has become increasingly popular because it links finite-horizon optimality with forward-invariance-based safety. Zeng et al. studied safety-critical MPC with discrete-time CBF constraints and showed how safety requirements can be incorporated directly into predictive optimization [34]. In maritime scenarios, Lee et al. recently proposed turning-circle-based CBFs for COLREGs-compliant collision avoidance, explicitly accounting for vessel maneuverability and avoidance direction [35]. These studies motivate the MPC-CBF foundation of the proposed safety filter.

Although CBF-based methods formalize safety, they primarily enforce invariance of a mathematically defined safe set. A CBF constraint may preserve a minimum distance, but it does not necessarily determine whether the own ship should act as a give-way, stand-on, or overtaking vessel. Thus, generic MPC-CBF can be geometrically safe while remaining difficult to justify under COLREGs. This gap motivates the integration of CBF safety constraints with COLREGs-aware semantic compliance envelope and runtime monitoring.

2.6. Runtime Assurance and Safety Filtering

Runtime assurance (RTA) is an architectural principle for systems that include complex, optimization-based, or learning-enabled components. Rather than requiring full offline validation of an advanced controller, RTA monitors runtime behavior and switches to, or filters through, a verified safety controller when necessary. The Simplex architecture introduced by Seto et al. is an early example of this idea, separating an advanced controller from a verified safety controller [36].

More recent RTA research extends this concept to learning-enabled autonomous systems. Cofer et al. demonstrated a runtime assurance architecture for a neural-network-based aircraft taxiing application, showing how monitors and high-assurance components can constrain the behavior of an advanced controller [37]. The same principle is applicable to ASV collision avoidance, which may involve predictive optimization, learned perception, or semantic instructions from LLMs. Such components may improve performance or interpretability, but their outputs should remain bounded by deterministic safety authority.

This literature motivates the RTA-MPC-CBF and single-envelope RTA-MPC-CBF baselines. The RTA-MPC-CBF baseline tests whether runtime safety supervision reduces worst-case behavior relative to nominal COLREGs-MPC-CBF. The single-envelope baseline further evaluates whether a fixed conservative safety envelope is sufficient, or whether

a dual-envelope design is needed to reduce unnecessary intervention during low-risk operation while retaining emergency protection.

2.7. Learning-Based and LLM-Assisted COLREGs Reasoning

Learning-based methods have been developed to improve flexibility in complex multi-vessel environments. Deep reinforcement learning (DRL), multi-agent reinforcement learning (MARL), and hybrid optimization-learning algorithms can learn collision-avoidance behavior from simulation and capture nonlinear relationships that are difficult to hand-code. Wei and Kuo proposed CA-QMIX, a COLREGs-compliant multi-ship collision-avoidance algorithm that integrates a 3-DOF ship model with ORCA-based collision-risk prediction and safe-velocity calculation [38]. Wen et al. proposed a two-stage DRL planning method for cooperative USV path planning under COLREGs constraints [39].

Learning-based approaches are promising because they can generalize across diverse scenarios and may learn maneuvers that are difficult to design manually. However, they also raise challenges related to interpretability, verification, distribution shift, and certification. A learned policy may perform well in simulation while providing limited explanation of its COLREGs reasoning. Because safety-critical maritime navigation requires auditable rule compliance and failure handling, learning-based components should be constrained by formal safety filters or placed in advisory roles rather than granted unrestricted control authority.

Recently, LLMs have attracted increasing attention as high-level decision-support tools because they can understand textual rules, provide human-readable reasoning, and translate complex situations into interpretable decisions. Agyei et al. proposed an LLM-based decision-making and control architecture for ASVs under COLREGs, in which the LLM serves as a high-level decision-maker and local planning and control modules implement the generated command [40]. Related interest is also reflected in the COLREG 3 report and the Navigation-GPT preprint, which explore LLMs for maritime rule interpretation and navigation support [41,42].

Nevertheless, using an LLM as a direct maneuver authority remains difficult to justify in safety-critical maritime control. LLM outputs may be unstable, overconfident, or semantically plausible yet geometrically unsafe. They may also depend strongly on prompt design, state representation, and scenario distribution. An LLM should therefore not be regarded as a certified controller. A safer strategy is to restrict the LLM to semantic governance: it may classify the encounter, identify the applicable COLREGs responsibility, suggest qualitative maneuver preferences, and provide an explanation, but its output must be structured, validated, spatially and temporally checked, and bypassed at runtime when necessary.

2.8. Research Gap and Position of This Work

The literature shows that COLREGs-compliant collision avoidance has evolved along several complementary directions. Rule-based, APF, VO, ORCA, and ship-domain methods provide interpretability and regulatory structure but often rely on manually encoded thresholds and encounter rules. MPC-based approaches improve predictive feasibility and dynamic constraint handling, but still depend on predefined COLREGs semantics. COLREGs-aware optimization improves the generation of rule-consistent trajectories, but it does not necessarily provide formal safety or runtime-failure handling. CBF and MPC-CBF methods provide formal safety filtering, but they do not directly resolve COLREGs semantic ambiguity. RTA methods can supervise untrusted advanced controllers, but require effective safety envelopes and do not themselves provide maritime rule interpretation. Learning-based and LLM-assisted methods improve adaptability and interpretability, but their outputs cannot be certified when treated as direct control commands.

This paper addresses this gap by proposing a runtime-assured dual-envelope predictive safety-control framework in which the LLM is not used as a direct controller. Instead, the LLM is confined to semantic governance, generating structured COLREGs interpretations, encounter classifications, and qualitative maneuver preferences. These outputs are then verified through schema validation, temporal-consistency checking, geometric-feasibility checking, confidence monitoring, and runtime assurance supervision. The deterministic MPC-CBF safety layer retains final command authority by enforcing collision-avoidance constraints and rejecting or downgrading unsafe semantic suggestions.

This review also motivates the baseline design in the experimental section. The COLREGs baseline represents hand-coded maritime rules. The APF/VO/ORCA baseline represents reactive geometric avoidance. The MPC-only baseline evaluates the benefit of finite-horizon prediction. The MPC-CBF baseline measures the effect of formal barrier safety. The COLREGs-MPC-CBF baseline evaluates rule-aware optimization. The RTA-MPC-CBF and single-envelope RTA-MPC-CBF baselines measure runtime safety supervision and the cost of conservative fixed-envelope protection. The proposed method further introduces semantic uncertainty awareness, adaptive safety inflation, and dual-envelope runtime assurance, thereby integrating semantic interpretability with deterministic safety enforcement.

3. Problem Formulation and Methodology

3.1. Design Principle and Overall Architecture

This paper proposes a runtime-assured dual-envelope predictive safety-control architecture for COLREGs-compliant ASVs. The main design principle is to decouple semantic rule interpretation from safety-critical control. The LLM is not used as a direct controller and is not permitted to output rudder angle, thrust, yaw rate, acceleration, waypoints, or other actuator-level commands. Instead, it serves as a semantic governor that maps structured encounter data to auditable COLREGs-related labels and qualitative maneuver preferences.

Final control authority remains with a deterministic safety layer. This layer combines a semantic compliance envelope, a predictive risk envelope, and an MPC-CBF safety filter. The semantic compliance envelope represents validated COLREGs interpretations and penalizes behavior inconsistent with the applicable rules. The predictive risk envelope represents physical collision risk by propagating target-vessel trajectories and prediction uncertainty over the prediction horizon. The MPC-CBF safety filter then computes the final admissible control command by imposing collision-avoidance constraints while minimizing unnecessary deviation from nominal path-following behavior and COLREGs-consistent preferences.

The overall runtime-assured dual-envelope architecture is shown in Figure 1. First, the state-estimation module provides own-ship and target-vessel states. These estimates are processed through two parallel channels: a semantic control channel and a predictive risk channel. The semantic branch converts deterministic encounter features into verified COLREGs labels and maps them to the semantic compliance envelope. In parallel, the prediction branch estimates target-vessel motion and uncertainty over the MPC horizon to construct the predictive risk envelope. The two envelopes are then combined by the MPC-CBF safety filter, while the runtime assurance supervisor monitors semantic validity, solver feasibility, and fallback conditions.

As illustrated in Figure 1, the LLM is not placed in the direct control loop because it does not make actuator-level decisions. Its output is converted into a semantically bounded envelope only after validation. This architectural separation allows semantic interpretations to guide the controller without becoming a single point of safety failure.

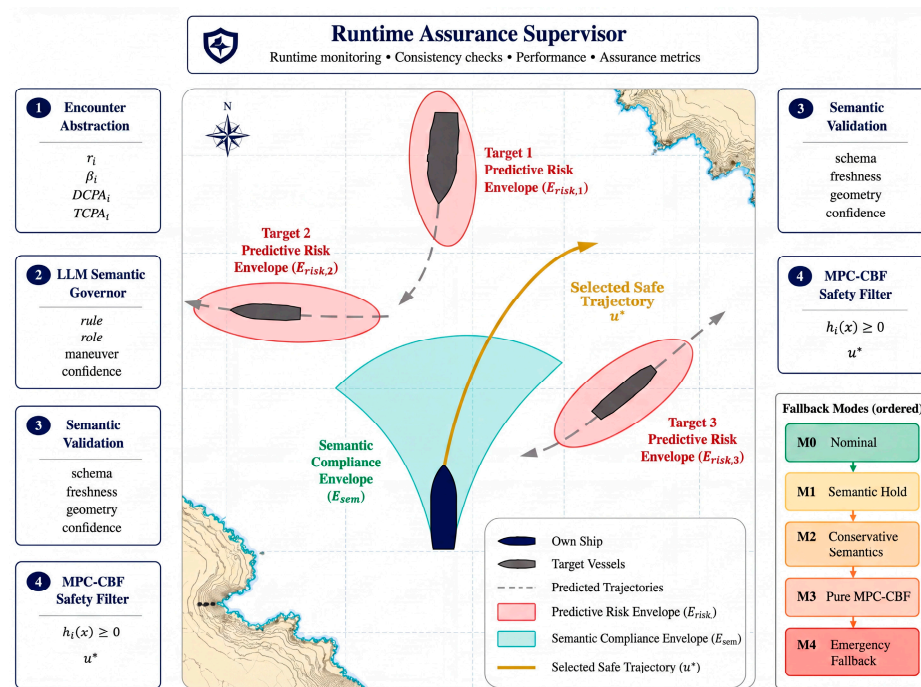


Figure 1. Overall architecture of the runtime-assured dual-envelope predictive safety-control framework. The LLM provides validated semantic governance only, whereas the MPC-CBF safety filter retains final command authority under runtime assurance supervision. In the figure, u^* denotes the optimized admissible control input generated by the MPC-CBF safety filter; the yellow curve illustrates the selected safe trajectory resulting from this optimized control action.

3.2. Vessel State Representation and Collision Risk Assessment

The controlled vessel is denoted as the own ship (OS). Surrounding vessels are indexed as target vessels TS_i , such that $i = 1, 2, \dots, N$. The state of the own ship and the state of the i -th target vessel are expressed as follows:

$$\eta_{OS} = [x_{OS}, y_{OS}, \psi_{OS}, U_{OS}]^T, \quad \eta_{TS_i} = [x_{TS_i}, y_{TS_i}, \psi_{TS_i}, U_{TS_i}]^T \quad (1)$$

where x and y denote vessel position, ψ denotes heading angle, and U is the vessel speed. With the x -axis pointing east and the y -axis pointing north, the velocity vectors of the own ship and the target vessel are given by the following:

$$v_{OS} = \begin{bmatrix} U_{OS} \sin \psi_{OS} \\ U_{OS} \cos \psi_{OS} \end{bmatrix}, \quad v_{TS_i} = \begin{bmatrix} U_{TS_i} \sin \psi_{TS_i} \\ U_{TS_i} \cos \psi_{TS_i} \end{bmatrix} \quad (2)$$

The relative position and velocity of target vessel i with respect to the own ship are defined as follows:

$$p_i = \begin{bmatrix} x_{TS_i} - x_{OS} \\ y_{TS_i} - y_{OS} \end{bmatrix}, \quad v_i = v_{TS_i} - v_{OS} \quad (3)$$

According to p_i and v_i , the time to closest point of approach and distance to closest point of approach are computed as follows:

$$TCPA_i = -\frac{p_i^T v_i}{v_i^T v_i}, \quad DCPA_i = |p_i + v_i TCPA_i|_2 \quad (4)$$

Because Equation (4) is singular and numerically unstable at relative velocities near zero (i.e., $|v_i| \rightarrow 0$), it is regularized by introducing a small regularization constant $\epsilon = 10^{-6}$ to the denominator. When $|v_i| < 0.05$ m/s, the system smoothly switches to the instant-

neous Euclidean distance $|p_i|_2$ as a conservative safety proxy, which reduces command chattering caused by discontinuities. When $TCPA_i < 0$, the vessels are considered receding from the closest point of approach and the encounter is not generally treated as an urgent one. A target vessel is considered dangerous when $0 < TCPA_i < T_{safe}$, $DCPA_i < D_{safe}$.

The normalized distance risk and time risk for target vessel i are mathematically formulated as follows:

$$\begin{aligned}\gamma_{D,i} &= \max\left(0, 1 - \frac{DCPA_i}{D_{safe}}\right) \\ \gamma_{T,i} &= \max\left(0, 1 - \frac{TCPA_i}{T_{safe}}\right)\end{aligned}\quad (5)$$

where D_{safe} and T_{safe} are the deterministic safety thresholds for closest-approach distance and time, respectively.

The overall risk index is

$$R_i = w_D \gamma_{D,i} + w_T \gamma_{T,i}, \quad w_D + w_T = 1 \quad (6)$$

The use of $w_D = 0.6$ and $w_T = 0.4$ as initial values in this study is because predicted passing distance directly indicates the severity of the possible conflict. Target vessels are ranked by R_i , $DCPA_i$, and $TCPA_i$. The highest-risk vessels are then included in the structured semantic prompt and predictive safety constraints. Including all surrounding vessels in the LLM prompt and the MPC optimization problem in high-density maritime traffic would cause substantial computational bottlenecks. To be scalable, the system uses a dynamic risk filter that utilizes the continuous risk index (R_i). Surrounding vessels are ranked by the risk index at each control step. Only the top $M = 3$ highest-risk vessels with $R_i \geq 0.30$ (medium-to-high risk) are active. The filtered vessels are used as constraints in the MPC-CBF safety filter and converted into a structured LLM description and the lower-risk vessels are temporarily ignored to maintain a bounded computational load. The threshold $D_{safe} = 5.0$ m is chosen to satisfy the collision boundary (approximately 2.5 times the vessel's length overall), ensuring a robust margin for maneuverability. T_{safe} is set to 30 s, which is consistent with typical observation intervals for small USVs to allow early intervention. The risk weights $w_D = 0.6$ and $w_T = 0.4$ are selected to prioritize predicted passing distance (spatial risk) over time-to-collision (temporal risk), reflecting a design preference for maintaining sufficient spatial clearance in the navigation corridor. The collision risk levels used for semantic encoding are summarized in Table 1.

Table 1. Collision risk levels used for semantic encoding.

Risk Level	Condition	Decision Implication
Safe	$DCPA_i \geq D_{safe}$ or $TCPA_i \leq 0$	No immediate avoidance action is required.
Low risk	$0 < R_i < 0.30$	The vessel is monitored but may not dominate the decision.
Medium risk	$0.30 \leq R_i < 0.60$	The vessel is included in semantic reasoning and safety verification.
High risk	$R_i \geq 0.60$ or $DCPA_i \ll D_{safe}$	The vessel is assigned high priority and must be considered by the safety filter.

3.3. COLREGs-Oriented Encounter Abstraction

Before querying the LLM, the numerical encounter state is transformed into a structured COLREGs-oriented representation. This step is essential because the LLM should not

infer vessel responsibilities directly from raw coordinates. Instead, it receives a rule-relevant context generated by deterministic encounter-classification logic.

$$\beta_i = (\theta_i - \psi_{OS} + \pi) \bmod 2\pi - \pi \quad (7)$$

where θ_i is the absolute bearing of the target vessel from the own ship, and ψ_{OS} is the own ship's heading. The modulo operation ensures the relative bearing β_i is mapped to $[-\pi, \pi]$.

Positive values indicate that the target vessel is on the starboard side of the own ship, whereas negative values indicate that it is on the port side. The course difference is defined as follows:

$$\Delta\psi_i = |(\psi_{TS_i} - \psi_{OS} + \pi) \bmod 2\pi - \pi| \quad (8)$$

The geometric quantities used for deterministic encounter abstraction are illustrated in Figure 2. Relative bearing, course difference, DCPA, and TCPA provide the numerical basis for identifying head-on, crossing, overtaking, and non-conflict situations before LLM semantic reasoning is performed.

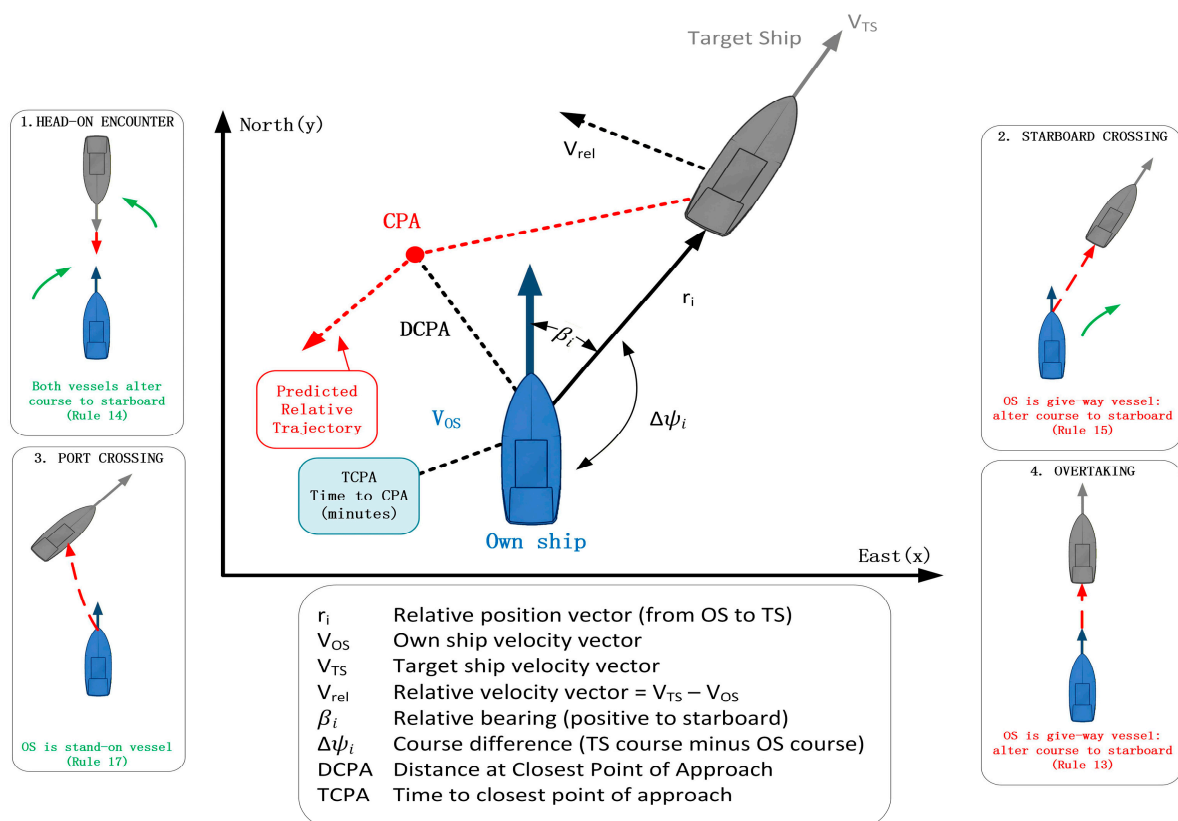


Figure 2. COLREGs-oriented encounter geometry used for deterministic abstraction. The own ship and target vessel are described by relative bearing, course difference, DCPA, TCPA, and predicted relative motion before semantic reasoning is performed.

Figure 2 emphasizes that the LLM does not infer vessel responsibility from raw coordinates. Instead, the encounter is first transformed into a bounded geometric representation. This design reduces ambiguity in LLM semantic reasoning and provides a reference for geometry-consistency validation.

Table 2 summarizes the rule-based encounter abstraction used before LLM semantic reasoning. These thresholds are used as initial simulation settings and can be adjusted according to vessel size, traffic density, sensor range, and operating area.

Table 2. Rule-based encounter abstraction used before LLM semantic governance.

Encounter Type	Classification Condition	Own-Ship Role	Preferred COLREGs-Consistent Behavior
Head-on	$ \beta_i \leq 10^\circ$ and $\Delta\psi_i \geq 150^\circ$	Both vessels give way	Alter course to starboard.
Starboard crossing	$10^\circ < \beta_i < 112.5^\circ$ and risk condition holds	Give-way vessel	Take early and substantial action; avoid crossing ahead.
Port crossing	$-112.5^\circ < \beta_i < -10^\circ$ and risk condition holds	Stand-on vessel	Maintain course and speed initially; monitor give-way action.
Overtaking	Own ship approaches target from the target's stern sector	Overtaking vessel gives way	Keep clear of the vessel being overtaken.
Non-conflict	Risk condition does not hold	No immediate obligation	Maintain course and speed.

The deterministic abstraction is not intended to replace COLREGs interpretation. Rather, it provides a bounded and repeatable input to the LLM. The LLM may help organize the semantics of uncertain or multi-vessel scenarios, but its outputs must remain consistent with the geometric encounter state.

3.4. LLM Semantic Governor and Structured Output Schema

This paper uses the GPT-4-Turbo model accessed through the OpenAI API as the foundation of the LLM semantic governor because of its strong natural-language understanding and instruction-following capability. To reduce hallucination and improve reproducibility, the prompt temperature is set to 0.1.

The LLM semantic governor is designed as an interpretable rule-reasoning module rather than a direct control module. At each semantic decision interval, the system converts the numerical encounter state into a condensed structured description, including target-vessel state, relative bearing, range, DCPA, TCPA, risk level, deterministic encounter abstraction, own-ship role, and candidate maneuver labels. This design provides the LLM with a constrained COLREGs context and prevents unconstrained inference about vessel responsibility based only on coordinates.

The LLM is allowed to output only semantic navigation labels. It cannot output rudder angle, thrust, yaw rate, acceleration, waypoints, or other actuator-level commands. The required semantic fields are listed in Table 3; they define the information that may be accepted from the LLM and transferred to the validation layer.

Table 3. Structured semantic output fields of the LLM governor.

Field	Description	Example Values
Encounter type	COLREGs-related encounter category	head-on, starboard crossing, port crossing, overtaking, multi-vessel, unclear
Own-ship role	Rule-based responsibility of the own ship	give-way, stand-on, overtaking, mixed, unclear
Applicable rules	Relevant COLREGs rules	Rule 8, Rule 13, Rule 14, Rule 15, Rule 16, Rule 17
Recommended maneuver	Qualitative maneuver preference	alter starboard, maintain course and speed, reduce speed, pass astern
Prohibited maneuvers	maneuvers that should be avoided	alter port across bow, increase speed toward target
Confidence	Self-reported semantic confidence	value between 0 and 1
Reasoning summary	Concise explanation for logging	short textual explanation
Fallback flag	Whether conservative fallback is required	true or false

Table 3 shows that the LLM output is restricted to a fixed semantic vocabulary. The output is therefore not treated as a navigation command but as a set of auditable semantic labels used to construct the soft semantic compliance envelope. Before these labels are allowed to influence the MPC-CBF safety filter, they are examined by the validation layer through schema, vocabulary, geometry, confidence, and temporal-consistency checks.

3.5. Semantic Validation Layer

Figure 3 summarizes the validation process for LLM-generated semantic labels. The structured LLM output is not sent directly to the controller; it must pass schema validation, vocabulary validation, geometry-consistency checking, confidence validation, and freshness validation.

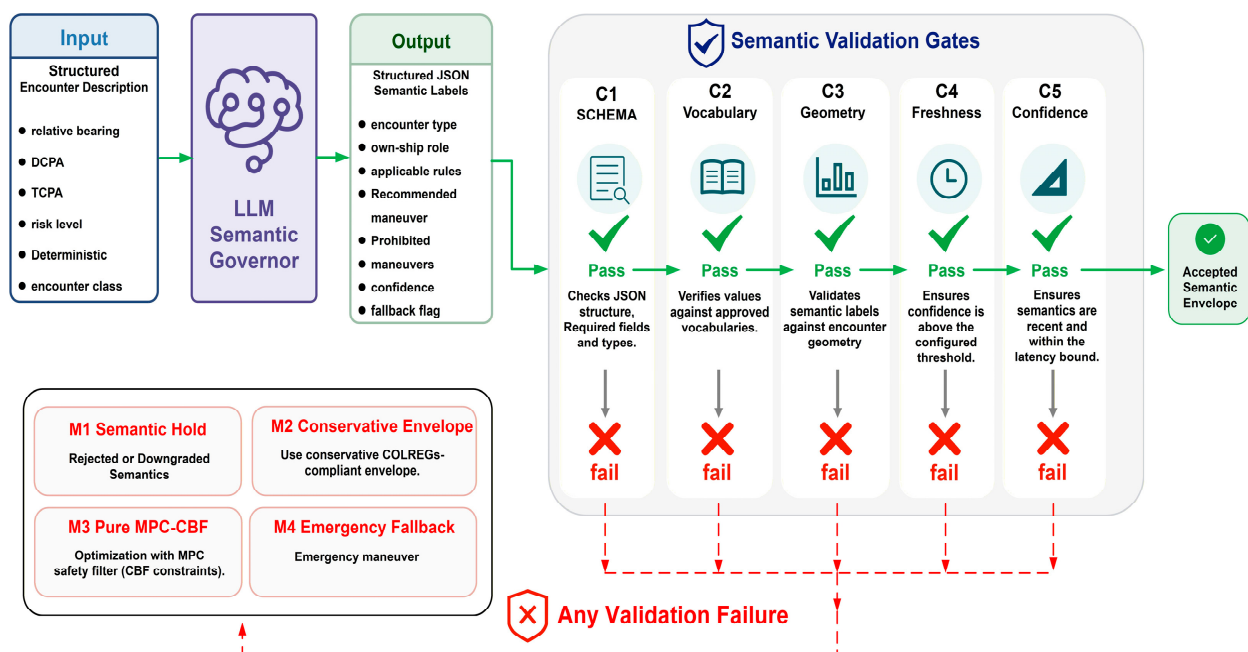


Figure 3. Validation pipeline for structured LLM semantic outputs. Malformed, low-confidence, stale, or geometrically inconsistent semantic labels are rejected or downgraded before they can influence the MPC-CBF safety filter.

The validation pipeline in Figure 3 is a key safety boundary of the proposed design. It ensures that semantically plausible but machine-unacceptable or geometrically inconsistent LLM outputs cannot influence the optimization problem. A failed validation stage causes the runtime assurance supervisor to select a fallback mode rather than treating the LLM output as trusted advice.

The semantic validation layer blocks untrusted language output from entering the control problem. Each LLM response is stamped with the encounter state used in the prompt, and the response is accepted only after passing the following five checks.

To begin with, schema validation checks whether the response is a valid JSON and contains all required fields. Second, vocabulary validation verifies that each field is part of the pre-established list of permitted labels. Third, geometry validation verifies whether the stated encounter type and own-ship role are consistent with the relative bearing, DCPA, TCPA, and deterministic encounter abstraction. Fourth, confidence validation checks whether the confidence value is higher than the minimum threshold $c_{\min}$. Fifth, temporal validation discards stale outputs whose age surpasses the semantic freshness threshold τ_{LLM} .

If any validation step fails, the runtime assurance supervisor prevents the LLM output from directly influencing the control problem. Depending on the failure mode and risk level, the supervisor may hold the last valid semantic compliance envelope, activate a conservative envelope, skip the semantic layer, or enter emergency mode. The validation checks for LLM-generated semantic outputs are summarized in Table 4.

Table 4. Validation checks for LLM-generated semantic outputs.

Check	Detection Criterion	Response
Schema validation	Invalid JSON or missing mandatory fields	Reject output.
Vocabulary validation	Field value outside allowed vocabulary	Reject output.
Geometry validation	Encounter label conflicts with bearing, DCPA, or TCPA	Reject or downgrade semantic authority.
Confidence validation	confidence < c_{min} or encounter type is unclear	Activate conservative semantic compliance envelope.
Temporal validation	Output age exceeds τ_{LLM}	Discard stale semantics.

When multiple LLM backbones are available, disagreement in own-ship obligation or encounter type is treated as semantic uncertainty rather than majority-vote truth. Under such disagreement, the supervisor demotes or bypasses the semantic compliance envelope.

3.6. Semantic Compliance Envelope

The semantic compliance envelope converts validated COLREGs semantics into soft optimization constraints and penalties. The envelope is not imposed as a hard constraint because collision avoidance must retain priority. Instead, slack variables represent semantic consistency and record when the controller deviates from the nominal COLREGs preference.

In a starboard-crossing encounter, the semantic compliance envelope for a give-way vessel penalizes passing ahead of the target vessel and favors starboard alteration, deceleration, or pass-astern geometry. In a port-crossing encounter, the semantic compliance envelope for a stand-on vessel favors maintaining course and speed unless the target vessel fails to give way or the predictive risk envelope indicates increasing danger. In an overtaking encounter, the envelope penalizes actions that fail to keep clear of the vessel being overtaken.

$$g_{\text{sem},j}(x_j, u_j; z_k) + \sigma_{\text{sem},j} \geq 0, \quad \sigma_{\text{sem},j} \geq 0 \quad (9)$$

where x_j is the predicted own-ship state at prediction step j , u_j is the corresponding control input, z_k is the validated semantic-label set at the current update step, and $\sigma_{\text{sem},j}$ is the semantic slack variable.

Semantic slack has two roles. Quantitatively, it keeps the optimizer feasible when semantic preferences conflict with urgent collision avoidance. Semantically, it serves as an audit signal: a large semantic slack value indicates that the controller had to depart from the nominal COLREGs preference to preserve physical safety. This is particularly important in stand-on situations, where the own ship should initially maintain course and speed but may need to maneuver if the give-way vessel does not respond. Thus, the semantic term guides rule-consistent behavior during normal operation, whereas the physical safety envelope and CBF constraints retain priority in safety-critical cases.

3.7. Predictive Risk Envelope

The predictive risk envelope is based on short-horizon target-vessel predictions. The prediction module provides the mean target position $\mu_{i,j}$ and covariance $\Sigma_{i,j}$ of each target vessel i and time step j . As the default predictor, constant-velocity or constant-turn-rate models can be considered. When sufficient trajectory data are available, a learned predictor

may be used as an advisory prediction module, but it should not be treated as a safety authority. Final safety enforcement remains with the MPC-CBF safety filter and the runtime assurance supervisor.

$$d_{\text{safe},i,j} = d_0 + k_\sigma \sqrt{\lambda_{\max}(\Sigma_{i,j})} + k_{\text{speed}} \|q_{i,j}\| \quad (10)$$

where d_0 is the base safety radius, k_σ is the uncertainty inflation coefficient, $\lambda_{\max}(\Sigma_{i,j})$ is the maximum eigenvalue of the predicted covariance matrix, and $q_{i,j}$ denotes the relative velocity-related risk term.

$$h_{i,j}(x_j) = \|p_{o,j} - \mu_{i,j}\|^2 - d_{\text{safe},i,j}^2 \quad (11)$$

where $p_{o,j}$ is the predicted own-ship position. The discrete-time CBF condition is written as follows:

$$h_{i,j+1}(x_{j+1}) - (1 - \alpha)h_{i,j}(x_j) + \sigma_{\text{safe},i,j} \geq 0 \quad (12)$$

where α is the CBF decay variable and $\sigma_{\text{safe},i,j}$ is the safety slack variable. Safety slack is retained only to preserve numerical feasibility and has a significantly greater penalty than semantic slack. Persistent activation of safety slack indicates that the controller, constraints, or scenario have moved into an unsafe or infeasible area.

3.8. MPC-CBF Safety Filter

Figure 4 shows how the semantic compliance envelope and predictive risk envelope are combined before entering the MPC-CBF safety filter. The semantic compliance envelope represents validated COLREGs-consistent maneuver preferences as soft penalties, whereas the predictive risk envelope represents uncertainty-inflated physical collision risk as safety constraints.

Figure 4 illustrates the distinct roles of the two envelopes. The predictive risk envelope defines the safety-critical region that the vessel should not enter, whereas the semantic compliance envelope represents the desired avoidance behavior derived from validated COLREGs semantics. This separation allows collision avoidance to override semantic preference when the two objectives conflict.

The safety filter of the MPC-CBF computes the latest admissible control command in a receding horizon. Nominal input $u_{\text{nom},j}$ can be provided by a path-following controller, line-of-sight guidance law, or waypoint tracker. The safety filter is not an alternative to nominal navigation. In the presence of risk or COLREGs-related constraints, it modifies the nominal command as little as possible.

With prediction horizon N , state x_j , and control input u_j , the safety-filter optimization problem is formulated as the safety-filter optimization problem as follows:

$$\begin{aligned} \min_{u_j, \sigma_{\text{sem},j}, \sigma_{\text{safe},i,j}} \quad & \sum_{j=0}^{N-1} \left(\|x_j - x_{\text{ref},j}\|_Q^2 + \|u_j - u_{\text{nom},j}\|_R^2 + \lambda_{\text{sem}} \sigma_{\text{sem},j}^2 + \lambda_{\text{safe}} \sum_i \sigma_{\text{safe},i,j}^2 \right) \\ \text{Subject to :} \quad & \begin{cases} x_{j+1} = f(x_j, u_j) \\ u_j \in \mathbb{U}, x_j \in \mathbb{X} \end{cases} \end{aligned} \quad (13)$$

Here, $x_{\text{ref},j}$ denotes the reference state, $u_{\text{nom},j}$ denotes the nominal control input, $\mathbb{U}$ is the admissible input set, and $\mathbb{X}$ is the admissible state set. The penalty weights satisfy $\lambda_{\text{safe}} \gg \lambda_{\text{sem}}$, ensuring that collision avoidance takes precedence over semantic preference when the two objectives conflict. The first feasible control input of the optimized sequence is applied to the ASV, and the optimization is repeated at the next MPC replanning step using the updated own-ship and target-vessel states. Table 5 defines the authority boundaries of the proposed control architecture.

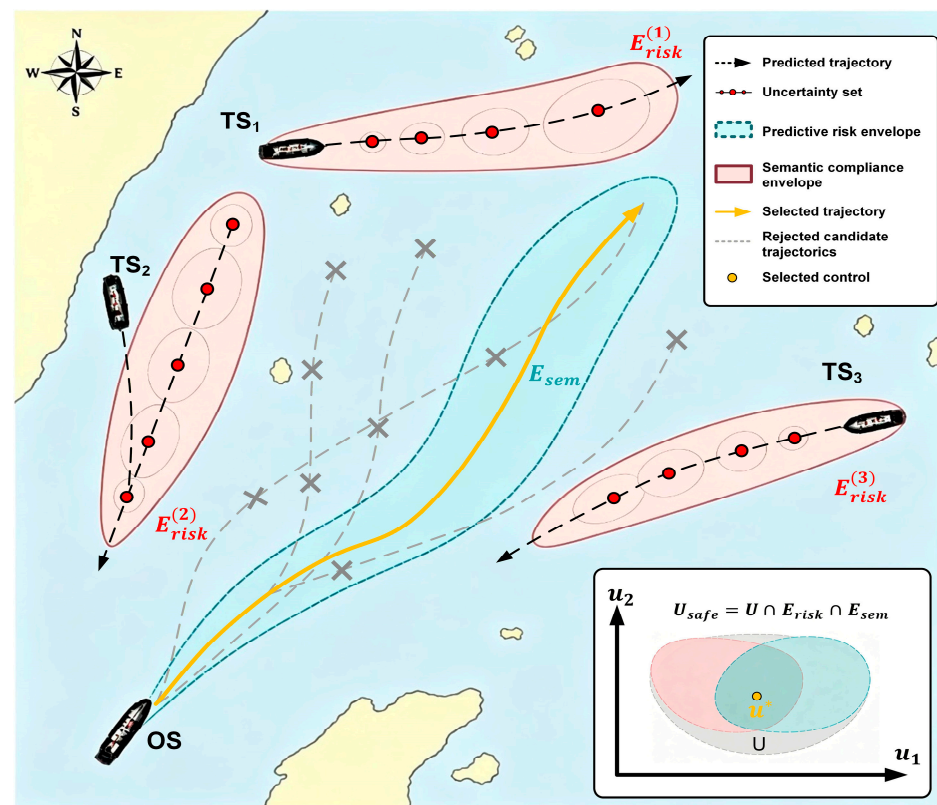


Figure 4. Fusion of the semantic compliance envelope and predictive risk envelope in the MPC-CBF safety filter. The semantic compliance envelope penalizes COLREGs-inconsistent behavior through slack variables, while the predictive risk envelope enforces physical collision-avoidance constraints. The dots along the target-vessel trajectories denote predicted target positions over the horizon, the crosses denote rejected candidate trajectories, the yellow curve denotes the selected safe trajectory, and the shaded regions denote the predictive risk envelope and the semantic compliance envelope.

Table 5. Authority boundaries of the proposed control architecture.

Component	Input	Output	Authority Boundary
Nominal path follower	Waypoints and own-ship state	Nominal command	No final authority when risk constraints are active.
LLM semantic governor	Structured encounter summary	Validated semantic labels	No actuator-level command authority.
Prediction module	Target-vessel state histories	Predicted means and covariances	No COLREGs interpretation authority.
MPC-CBF safety filter	Nominal command, semantic compliance envelope, risk envelope	Final command	Final normal-operation command authority.
Runtime assurance supervisor	Validity, latency, risk, solver status	Assurance mode and fallback selection	Can bypass the semantic layer or trigger emergency behavior.

3.9. Runtime Assurance Supervisor

The runtime assurance supervisor manages the interaction between the semantic layer and the deterministic control layer. It operates through discrete assurance modes and switches modes according to LLM validity, semantic uncertainty, predictive risk, and solver status.

Nominal mode uses both the validated semantic compliance envelope and the predictive risk envelope. Semantic-hold mode reuses the most recent valid semantic compliance

envelope when LLM output is temporarily unavailable. Conservative-semantic mode applies a predefined conservative envelope under low LLM confidence or ambiguous encounters. Pure-MPC-CBF mode removes the semantic compliance envelope when LLM output is malformed, geometrically inconsistent, or persistently stale. Emergency mode is activated when the optimizer becomes infeasible, safety slack grows excessively, or collision risk cannot be reduced by ordinary commands.

In emergency fallback mode (M4), the ASV does not simply cut propulsion because drifting in environmental disturbances may lead to loss of steerage. Instead, the supervisor executes a safe-harbor dynamic maneuver: it commands a full-rudder turn to starboard, consistent with COLREGs Rule 2, or initiates a dynamic-positioning braking sequence. This strategy dissipates kinetic energy while preserving heading control against wind and currents.

Figure 5 shows the mode transitions of the runtime assurance supervisor. The supervisor switches among nominal mode, semantic-hold mode, conservative-semantic mode, pure MPC-CBF mode, and emergency fallback mode according to LLM validity, semantic freshness, geometric consistency, solver feasibility, and safety-slack activation.

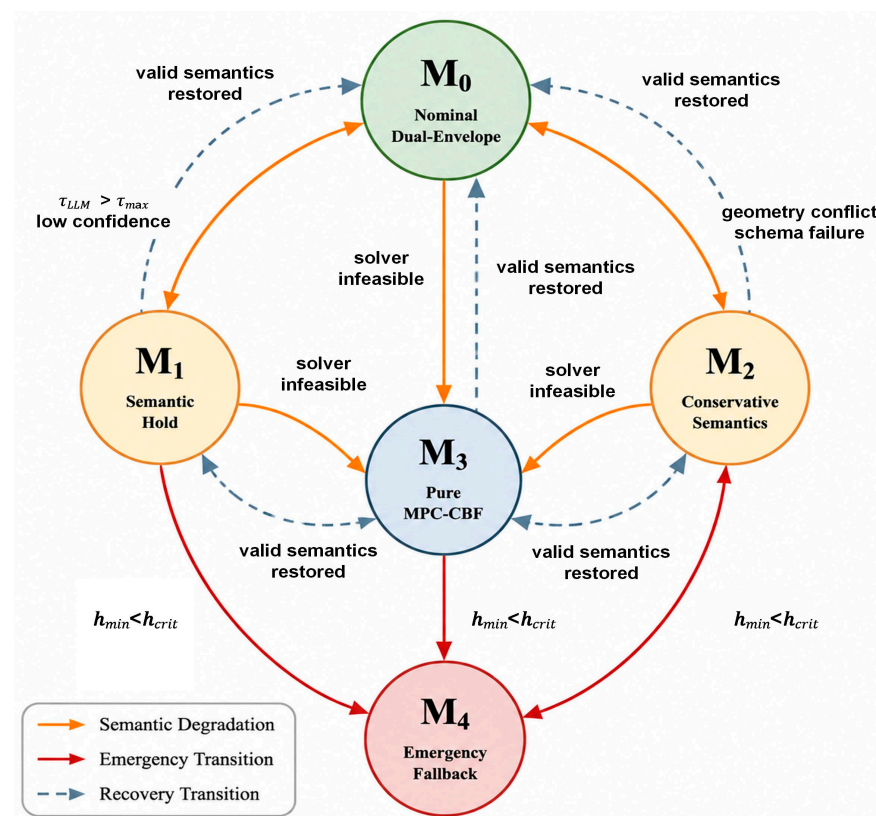


Figure 5. Runtime assurance mode transitions under semantic and optimization failures. The supervisor can downgrade or bypass the semantic layer when LLM outputs become unreliable, and it can trigger emergency fallback when solver feasibility or physical safety is threatened.

As illustrated in Figure 5, fallback activation is not treated as a software anomaly but as an anticipated safety response. When the LLM output is delayed, malformed, low-confidence, or geometrically inconsistent, the supervisor prevents it from directly affecting the control command and shifts the system to a safer operating mode.

The runtime assurance supervisor monitors the validity, freshness, consistency, and safety influence of both the semantic layer and the MPC-CBF solver. When the LLM output is unreliable, the supervisor prevents it from influencing the control problem and either retains the last valid semantic compliance envelope, activates a conservative envelope,

bypasses the semantic layer, or enters emergency safety mode. The monitoring logic and fallback actions are summarized in Table 6. Algorithm 1 summarizes the runtime-assured dual-envelope COLREGs control loop.

Algorithm 1. Runtime-assured dual-envelope COLREGs control loop.

Input:

Own-ship state $x_o(k)$, target-vessel states $X_T(k)$, nominal reference $r(k)$,
COLREGs rule set R , LLM semantic governor G , prediction horizon N ,
control period Δt , LLM query interval T_{LLM} , semantic freshness threshold τ_{LLM} .

Output:

Safe control command $u^*(k)$.

1: Initialize the runtime-assurance supervisor S_{RA} .

2: Initialize the semantic compliance envelope as $E_{sem} \leftarrow E_{cons}$.

3: Set the last accepted LLM update time as $t_{LLM} \leftarrow -\infty$.

4: Set the asynchronous LLM query handle as $q_{LLM} \leftarrow \emptyset$.

5: Let $K_{LLM} = \text{round}(\frac{T_{LLM}}{\Delta t})$.

6: while the ASV mission is active do

7: Acquire $x_o(k)$ and $X_T(k)$.

8: Compute range, relative bearing, DCPA, TCPA, encounter class, and risk level.

9: Rank target vessels and construct the high-risk target set $H(k)$.

10: Construct the structured encounter description $z(k)$.

11: if $k \bmod K_{LLM} = 0$ and $q_{LLM} = \emptyset$ then

12: Launch an asynchronous LLM query:

$q_{LLM} \leftarrow G(z(k), H(k), R)$.

13: end if

14: if q_{LLM} has returned a new semantic output y_{LLM} then

15: Validate y_{LLM} using schema, vocabulary, confidence,
 geometric-consistency, and freshness checks.

16: if y_{LLM} is valid then

17: $E_{sem} \leftarrow \text{MapToSemanticEnvelope}(y_{LLM})$.

18: $t_{LLM} \leftarrow k\Delta t$.

19: Set S_{RA} to nominal semantic-governance mode.

20: else

21: Reject y_{LLM} .

22: $E_{sem} \leftarrow \text{SemanticFallback}(E_{sem}, E_{cons}, S_{RA})$.

23: end if

24: $q_{LLM} \leftarrow \emptyset$.

25: end if

26: if $k\Delta t - t_{LLM} > \tau_{LLM}$ then

27: Mark the semantic output as stale.

28: $E_{sem} \leftarrow \text{SemanticFallback}(E_{sem}, E_{cons}, S_{RA})$.

29: end if

30: Predict target-vessel trajectories over the MPC horizon.

Algorithm 1. *Cont.*

```

31: Construct the predictive risk envelope with uncertainty inflation.
32: Formulate the dual-envelope MPC-CBF optimization problem using  $E_{sem}$ .
33: Solve the MPC-CBF optimization problem.

34: if the optimization is feasible and the solver time is acceptable then
35:     Apply the first optimal control input:
         $\mu^*(k) \leftarrow u_0^*$ .
36: else
37:     Activate emergency runtime-assurance mode.
38:     Apply the predefined safe fallback command:
         $\mu^*(k) \leftarrow u_{safe}$ .
39: end if

40: Log semantic labels,  $E_{sem}$ , slack variables, CPA, CBF margin,
    fallback mode, and solver time.
41: end while

```

Table 6. Runtime assurance triggers and fallback actions.

Monitored Item	Failure Condition	Supervisor Decision	Fallback Action	Purpose
LLM output format	The output is not valid JSON, or required fields are missing.	Reject the LLM output.	Use the last valid semantic compliance envelope or conservative envelope.	Prevent malformed language output from entering the controller.
LLM vocabulary	The output contains labels outside the predefined vocabulary.	Reject the LLM output.	Use conservative semantic guidance.	Ensure that the semantic layer remains machine-readable and bounded.
Geometric consistency	The LLM encounter type conflicts with relative bearing, DCPA, TCPA, or deterministic encounter abstraction.	Downgrade or reject semantic authority.	Rely on the predictive risk envelope and MPC-CBF safety filter.	Prevent semantically plausible but geometrically wrong decisions.
Confidence level	The LLM confidence is lower than the threshold c_{min} .	Downgrade semantic authority.	Activate conservative semantic compliance envelope.	Avoid over-trusting uncertain semantic reasoning.
Output freshness	The LLM output age exceeds τ_{LLM} .	Discard stale semantic output.	Hold the last valid semantic compliance envelope for a limited time, then bypass the LLM.	Prevent delayed semantic decisions from affecting current control.
Solver feasibility	The MPC-CBF problem becomes infeasible or safety slack remains active.	Trigger safety-priority mode.	Reduce speed, enlarge safety margin, or enter emergency avoidance mode.	Preserve collision avoidance when the optimization problem becomes unsafe.

The framework addresses the natural time-scale mismatch between the 10 Hz deterministic command-publishing layer, the 2 Hz MPC-CBF replanning task, and the asynchronous LLM semantic governor. At each 100 ms command tick, the RTA supervisor publishes the newest validated MPC-CBF command, a cached safe command, or an emergency fallback command. The MPC-CBF optimizer runs at a 500 ms replanning period and is terminated by a 480 ms watchdog if it does not return in time. The LLM semantic

governor runs asynchronously and updates the cached semantic compliance envelope only after validation.

4. Experiments and Results Analysis

4.1. Experimental Objectives and Contribution-to-Experiment Mapping

To provide the physical simulation with an engineering background, the simulated autonomous surface vehicle is represented as scaled to a typical survey USV (e.g., like the Maritime Otter USV platform), having a length overall (LOA) of $L = 2.0$ m and beam of $B = 1.05$ m. The base safety radius is therefore $d_0 = 3.0$ m, and the minimum physical collision boundary is $d_{min} = 5.0$ m (about 2.5 times the LOA) which provides a physically reasonable safety margin when the environment is disturbed.

The experiments are designed to answer the six research questions related to the main contributions. Firstly, is MPC more effective in anticipatory behavior compared to reactive or rule-based approaches? Secondly, are CBF limitations less prone to safety-margin infringements than unconstrained MPC? Thirdly, does COLREGs-aware encoding minimize rule-inconsistent maneuvers over generic collision-avoidance MPC-CBF? Fourthly, is runtime assurance able to prevent rare but catastrophic failures when the nominal optimizer fails or is delayed? Fifthly, can a dual-envelope mechanism minimize unnecessary emergency intervention over a single conservative envelope? Sixthly, can semantic uncertainty awareness enhance robustness in the case of ambiguous target-vessel behavior without making the controller excessively conservative in clear encounters? Table 7 maps the proposed technical contributions to the required validation experiments.

Table 7. Mapping from new technical contributions to required validation experiments.

New Technical Element	Why It Must Be Tested	Direct Comparison/Ablation	Main Evidence to Report
MPC planner	Shows whether finite-horizon prediction improves early avoidance rather than reacting late.	MPC vs. rule-based COLREGs, APF, VO/ORCA if available	CPA, intervention time, path penalty, maneuver smoothness
CBF safety constraints	Shows whether formal safety margins reduce unsafe trajectories generated by plain MPC.	MPC-CBF vs. MPC-only	Minimum CBF margin, collision count, infeasibility rate
COLREGs-aware MPC-CBF	Shows whether encoded maritime rules reduce technically safe but rule-inconsistent behavior.	COLREGs-MPC-CBF vs. generic MPC-CBF	Rule-violation count, stand-on deviation, give-way timing
Runtime assurance	Shows whether the system remains safe when the nominal planner fails, delays, or violates the envelope.	RTA-MPC-CBF vs. COLREGs-MPC-CBF without safety filter	Fallback activations, emergency violations, worst-case CPA
Dual-envelope design	Shows whether normal and emergency envelopes balance safety and efficiency better than one fixed envelope.	Dual-envelope RTA vs. single-envelope RTA	Fallback ratio, path increase, ETA increase, oscillation count
Semantic uncertainty module	Shows whether confidence-aware decisions improve ambiguous and noisy encounters.	Full method vs. no-semantic/no-uncertainty variants	Robust CPA, false intervention rate, uncertainty-stress success rate
Adaptive safety inflation	Shows whether the controller is conservative only when uncertainty requires it.	Adaptive inflation vs. fixed inflation	Safety under noise, path penalty in clear encounters, envelope-radius profile

4.2. Experimental Platform and Common Protocol

4.2.1. Simulation and Vessel Dynamics

All methods are evaluated in the same closed-loop simulation framework. The own ship obtains target-vessel states, classifies the encounter, predicts the future motion of nearby vessels, computes a nominal control action, and implements it after safety verification. The vessel model has three degrees of freedom in the horizontal plane, including surge, sway, and yaw. Current and wave-induced perturbations are introduced as environmental disturbances so that the controller is not tested only in an ideal deterministic environment.

Each baseline and ablation use the same ship model, actuator limits, prediction horizon, sampling time, traffic scenarios, initial conditions, and random seeds. This protocol is essential because the proposed method adds several modules; otherwise, observed differences could reflect changed experimental conditions rather than the algorithm itself.

Wind and ocean currents are represented as bounded, unmodeled dynamics and added to the vessel's kinetic equations. Robustness to these disturbances is addressed through the uncertainty inflation coefficient in the predictive risk envelope.

4.2.2. Hardware and Software Configuration

Computational-time reproducibility and system stability are assessed using a single platform. The hardware configuration consists of an Intel Core i9-13900K processor (5.8 GHz), 64 GB of DDR5 RAM, and an NVIDIA RTX 4090 graphics card with 24 GB of VRAM. Environment rendering and lightweight perception processing are accelerated on the GPU, whereas MPC-CBF optimization is executed on the CPU to characterize worst-case control latency.

The software environment is based on Ubuntu 22.04 LTS. Vessel dynamics, environment simulations, and COLREGs baseline scenarios are implemented in Python 3.10 and Robot Operating System 2 (ROS 2 Humble). Nonlinear MPC and CBF constraints are formulated using CasADi (v3.5.5) and solved with the IPOPT nonlinear programming solver (v3.14.4). The semantic-governance layer uses asynchronous LLM inference through the OpenAI API (GPT-4-Turbo-1106-preview) to decouple the high-frequency 10 Hz control loop from network latency. The 10 Hz rate refers to the deterministic RTA command-publishing layer rather than the nonlinear MPC-CBF solve rate. At each 100 ms command tick, the RTA layer publishes the newest validated MPC-CBF command, a cached safe command, or an emergency fallback command; it does not wait synchronously for a new MPC solution at every tick.

4.2.3. Fair Comparison Protocol and Baselines

To strictly assess the incremental contribution of each of the proposed modules and provide a statistically valid comparison, we define a rigorous assessment protocol using randomized Monte Carlo simulations. Each of the seven test scenarios (S1–S7) is run with every baseline and the proposed method on 100 randomized trials. On every trial, the initial states of both the own ship (OS) and the target vessels (TS) are perturbed. Specifically, the initial positions are perturbed with a uniform distribution of ± 15 m, headings with $\pm 8^\circ$, and initial speeds with $\pm 10\%$. The state measurements are perturbed with additive Gaussian noise to reproduce realistic maritime perception constraints with standard deviations of $\sigma_{pos} = 0.5$ m and $\sigma_{vel} = 0.1$ m/s on position and velocity, respectively.

To ensure a fair comparison, all approaches were tested using the same randomized seeds so that each of them encounters the same environmental disturbances and target maneuvers. All optimization-based planners (B2 to Ours) have a uniform prediction horizon of $N = 30$ steps with a sampling interval of $\Delta t = 0.5$ s giving a look-ahead horizon of 15 s. The baselines are designed progressively as indicated in Table 8:

1. B0 (Rule-based COLREGs): a deterministic lookup-table baseline that applies preset rudder commands according to geometric encounter sectors.
2. B1 (APF/VO): a reactive geometric collision-avoidance baseline that uses artificial potential fields or velocity obstacles to generate steering corrections.
3. B2 (MPC-only): a basic model predictive controller that optimizes path following and nominal collision avoidance without formal CBF safety constraints.
4. B3 (MPC-CBF): a controller that introduces control barrier functions into the MPC problem to enforce hard safety margins, but does not include semantic COLREGs rules.
5. B4 (COLREGs-MPC-CBF): a controller that embeds static COLREGs constraints into the MPC-CBF objective without runtime adaptation.
6. B5 (RTA-MPC-CBF): a controller that adds runtime assurance and switches to fallback mode under solver failure while using a fixed safety envelope.
7. B6 (Single-envelope RTA): a controller that uses a single highly conservative safety envelope with RTA supervision to maintain safety at all times.

Table 8. Scenario suite and the main module evaluated in each scenario.

Scenario	Description	Primary Difficulty	Primary Module Evaluated	Expected Behavior
S1 Head-on	Two vessels approach nearly reciprocal courses.	Symmetric collision risk and early action.	COLREGs-aware MPC + CBF	Both vessels or own ship alter to starboard with positive CBF margin.
S2 Crossing: own ship give-way	Target is on the starboard side of own ship.	Early and substantial give-way action.	MPC prediction + COLREGs constraints	Own ship changes course or speed early enough to pass safely.
S3 Crossing: own ship stand-on	Own ship has stand-on role but target may act late.	Avoiding unnecessary overreaction while retaining safety.	Runtime assurance + dual envelope	Maintain course if safe; intervene only when risk becomes critical.
S4 Overtaking	Own ship or target overtakes from abaft the beam.	Maintaining safe passing and avoiding ambiguous role changes.	Semantic encounter classification	Overtaking vessel keeps clear with smooth return to route.
S5 Multi-vessel mixed encounter	Several targets create competing constraints.	Constraint conflicts and possible optimizer infeasibility.	Runtime assurance + solver robustness	Prioritize highest-risk targets without oscillatory switching.
S6 AIS replay dense traffic	Recorded or replayed realistic traffic geometry.	Clutter, non-symmetric interactions, non-ideal target behavior.	Full integrated architecture	Complete route with low rule-violation and acceptable path penalty.
S7 Noisy perception and uncertain intention	Target states include noise and ambiguous intention changes.	Semantic uncertainty and confidence degradation.	Semantic uncertainty + adaptive safety inflation	Increase safety envelope under uncertainty; avoid excessive detour when confidence returns.

A trial is considered successful when all three of the following conditions are satisfied throughout the run:

1. The minimum physical separation between the own ship and any target vessel i must exceed the collision boundary, i.e., $\min_t \|p_t - p_i\|_2 \geq d_{\min}$ (where $d_{\min} = 5.0$ m in this study).
2. The own ship reaches the target waypoint within the maximum time limit of 200 s.

3. Control stability is maintained, meaning that unrecoverable solver timeouts and actuator-command losses do not occur, and all primary solver infeasibilities are handled by the RTA supervisor.

If any of these conditions is not satisfied, the trial is classified as a failure, such as a collision, deadlock, or computational crash, and is counted toward the failure rate.

4.2.4. Simulation and Test Scenarios

The scenario suite includes canonical two-vessel COLREGs problems, multi-vessel mixed encounters, realistic traffic replay, and uncertainty-stress cases. Canonical scenarios test whether the controller behaves correctly under interpretable rules, whereas dense and uncertain scenarios evaluate whether the added MPC, runtime assurance, and semantic-uncertainty modules provide practical benefit beyond demonstration.

4.2.5. Evaluation Metrics

The metrics are selected to avoid a common weakness in collision-avoidance evaluation: reporting only collision-free completion. A controller may avoid collision by generating unrealistically circuitous routes, violating COLREGs, or continuously activating emergency overrides. Therefore, safety, rule compliance, efficiency, smoothness, robustness, and runtime are reported jointly. The metrics used for all baselines and ablations are listed in Table 9.

Table 9. Metrics used for all baselines and ablations.

Metric Group	Metric	Meaning	Desired Direction
Safety	Minimum CPA	Minimum closest point of approach over one run.	Higher
Safety	Minimum CBF margin	Minimum value of the active barrier function.	Non-negative/higher
Safety	Collision or near-miss count	Number of runs below the required safety distance.	Lower
COLREGs	Rule-violation count	Detected violations of applicable head-on, crossing, or overtaking behavior.	Lower
COLREGs	Stand-on deviation	Unnecessary heading/speed deviation when own ship is stand-on.	Lower
Efficiency	Path-length increase	Relative increase compared with the nominal route.	Lower
Efficiency	ETA increase	Arrival-time penalty compared with no-encounter route.	Lower
Smoothness	Rudder-rate or heading-rate cost	Aggressiveness and oscillation of maneuvers.	Lower
Runtime assurance	Fallback ratio	Percentage of control cycles overridden by the safety layer.	Lower, if safety remains satisfied
Runtime	Mean/95th/max solver time	Online computational feasibility and worst-case delay.	below the MPC replanning deadline or handled by RTA watchdog
Robustness	Uncertainty-stress success rate	Success under noisy observations and ambiguous intention.	Higher

4.3. Baseline Methods

To evaluate the proposed architecture systematically, we define a progressive sequence of baseline methods (B0–B6). This structure isolates the contribution of each added component,

from reactive potential fields to runtime-assured control, so that performance improvements can be interpreted and attributed to the proposed dual-envelope modules. For brevity, quantitative results for all randomized Monte Carlo trials are summarized in Table 10.

Table 10. Baseline hierarchy for evaluating the proposed method.

Label	Method	Included Modules	Excluded Modules
B0	Rule-based COLREGs	Discrete COLREGs rules and fixed safety threshold.	MPC, CBF, runtime assurance, semantic uncertainty.
B1	APF or VO/ORCA	Reactive geometric collision avoidance.	Formal COLREGs encoding, CBF, runtime assurance.
B2	MPC-only	Finite-horizon optimization and actuator constraints.	CBF, runtime assurance, semantic uncertainty.
B3	MPC-CBF	MPC plus CBF safety constraints.	Explicit COLREGs role logic, runtime assurance, semantic uncertainty.
B4	COLREGs-MPC-CBF	MPC, CBF, and COLREGs role constraints.	Runtime assurance and semantic uncertainty.
B5	RTA-MPC-CBF	COLREGs-MPC-CBF plus runtime safety filter.	Semantic uncertainty and adaptive inflation.
B6	Single-envelope RTA-MPC-CBF	Runtime assurance with one conservative envelope.	Dual-envelope switching.
Ours	SU-RTA-MPC-CBF	MPC, CBF, COLREGs constraints, runtime assurance, dual envelope, semantic uncertainty, adaptive inflation.	None.

4.4. Main Comparative Experiments

4.4.1. Overall Comparison Across All Scenarios

The main comparison evaluates whether the complete proposed approach improves safety and compliance while maintaining reasonable efficiency and real-time feasibility.

As shown in Table 11, the proposed SU-RA-MPC-CBF method does not achieve the highest absolute success rate among all baselines. The single-envelope RA baseline obtains the highest success rate of 91.2% and the largest minimum CPA of 22.5 m, indicating that a highly conservative safety envelope can improve physical separation. However, this improvement is achieved at the cost of a substantial path-length penalty of 38.4% and an ETA penalty of 42.5%, which may be undesirable for practical navigation.

Table 11. Overall comparison of baselines and the proposed method.

Method	Success Rate (%)	Min. CPA (m)	CBF Violation Rate (%)	COLREGs Violations/Run	Path + (%)	ETA + (%)	Mean Solver Time (ms)
B0 Rule-based	68.4 ± 8.5	8.4 ± 6.2	N/A	1.85 ± 0.8	6.5 ± 3.2	8.1 ± 3.5	14.5 ± 4.2
B1 APF/VO	74.2 ± 7.6	10.2 ± 5.8	N/A	2.15 ± 0.9	9.2 ± 3.8	10.5 ± 4.1	18.2 ± 5.5
B2 MPC-only	80.5 ± 6.4	6.5 ± 8.5	22.4 ± 8.6	1.65 ± 0.7	5.8 ± 2.4	6.4 ± 2.6	85.6 ± 18.4
B3 MPC-CBF	84.6 ± 5.2	12.8 ± 5.4	4.5 ± 2.8	1.88 ± 0.8	11.4 ± 4.1	12.8 ± 4.5	112.4 ± 24.5
B4 COLREGs-MPC-CBF	86.2 ± 4.8	13.5 ± 4.8	3.8 ± 2.5	0.84 ± 0.4	14.2 ± 4.6	15.6 ± 4.8	135.8 ± 32.2
B5 RTA-MPC-CBF	87.5 ± 4.4	14.2 ± 4.5	2.5 ± 1.8	0.75 ± 0.3	16.5 ± 5.2	18.2 ± 5.5	142.5 ± 35.6
B6 Single-envelope	91.2 ± 3.5	22.5 ± 5.6	0.8 ± 0.5	0.65 ± 0.3	38.4 ± 12.5	42.5 ± 14.8	158.6 ± 42.5
Ours SU-RTA-MPC-CBF	89.5 ± 4.2	16.8 ± 4.8	1.2 ± 0.8	0.42 ± 0.2	16.5 ± 5.4	17.8 ± 6.2	165.2 ± 48.4

In comparison, the proposed dual-envelope method achieves a slightly lower success rate of 89.5%, but it reduces the path-length penalty to 16.5% and the ETA penalty to

17.8%. These results suggest that the proposed method provides a more balanced compromise between safety, COLREGs compliance, and navigation efficiency, rather than simply maximizing conservatism. Therefore, the main advantage of the proposed method lies in improving the safety–efficiency trade-off under semantic uncertainty.

4.4.2. Scenario-Level Comparison of the Full Method

Scenario-level analysis prevents the overall average from masking weaknesses in specific COLREGs cases. For example, a controller may perform well in head-on scenarios but poorly in stand-on crossing situations if it reacts excessively and violates expected stand-on behavior. Therefore, each scenario is evaluated separately using the full method. The scenario-level results of the proposed method are presented in Table 12.

Table 12. Scenario-level results of the proposed method.

Scenario	Success Rate (%)	Min. CPA (m)	Min. CBF Margin	COLREGs Violations/Run	Fallback Ratio (%)	Path + (%)	Max Solver Time (ms)
S1 Head-on	98.5	16.4	0.85	0.08	4.5	10.2	165
S2 Crossing give-way	96.2	14.8	0.62	0.15	8.2	12.5	182
S3 Crossing stand-on	95.8	12.5	0.45	0.25	15.6	8.4	215
S4 Overtaking	97.4	18.2	0.95	0.05	3.8	9.5	158
S5 Multi-vessel	78.5	8.4	−0.15	1.45	42.5	24.6	385
S6 AIS replay	74.2	6.5	−0.28	1.82	55.4	28.5	425
S7 Uncertainty stress	81.6	10.2	0.12	0.85	62.8	22.4	394

The scenario-level results in Table 12 further show that the proposed method performs well in canonical two-vessel encounters, including head-on, crossing, and overtaking scenarios. In S1–S4, the success rates remain above 95%, the minimum CBF margins are positive, and the COLREGs violation rates are relatively low. This indicates that the proposed semantic-RTA framework is effective when the encounter geometry is structurally clear and the COLREGs role assignment is not strongly conflicting.

However, the dense multi-vessel scenario (S5) and AIS replay scenario (S6) expose the main limitation of the current implementation. In these two cases, the minimum CBF margins become negative (−0.15 and −0.28, respectively), while the fallback ratios increase to 42.5% and 55.4%. These negative CBF margins should not be interpreted as continuous violation of the safety constraint throughout the entire trajectory. Instead, they indicate short-duration constraint violations or slack-activated relaxations that occur during highly congested or semantically conflicting encounters. In such cases, the runtime assurance layer prevents prolonged open-loop execution and switches to fallback commands, but it cannot fully eliminate all transient constraint violations under severe multi-vessel conflicts.

Therefore, the proposed method should be interpreted as improving robustness and reducing the severity of unsafe behavior in dense scenarios, rather than guaranteeing strict CBF feasibility under all operating conditions. This limitation motivates further work on multi-vessel priority assignment, less conservative yet formally bounded slack design, and scalable optimization for dense maritime traffic. The spatial trajectories in Figure 6 are consistent with the statistical results reported in Tables 11 and 12. The proposed approach manages shrinking geometric gaps by temporarily relaxing COLREGs semantic preferences, thereby avoiding the extreme path penalty typically associated with rigid conservative envelopes. It is important to note that the observed performance degradation in these dense scenarios is partly attributed to the inherent limitations of our current simulation model. Specifically,

the discrepancy between the simplified vessel dynamics used in the optimizer and the more complex reality of maritime environmental perturbations (e.g., non-uniform wind and current disturbances) makes the ‘worst-case’ state more difficult to predict. Furthermore, the reliance on target-motion prediction, which assumes constant-velocity or constant-turn-rate behavior, can lead to under-inflation of the risk envelope in scenarios with highly non-linear target maneuvers. These factors, combined with the computational latency induced by dense constraints, contribute to the frequent trigger of the runtime assurance fallback mechanism. This underscores that while the proposed framework enhances robustness, it remains constrained by the fidelity of the underlying prediction and dynamics models.

Quantitative comparison in multi-ship game scenarios ($n = 50$).

Method	Success Rate (%)	Collision Rate (%)	Mean Time to Goal (min)	Mean Path Length (NM)	Path Length Penalty (%) (↓ is better)	Mean Min. Distance (NM)	Computational Time (s)
MPC-only (B2)	96.0	4.0	70.3 ± 18.5	108.9 ± 24.6	+3.2%	0.08 ± 0.05	0.12 ± 0.03
Single-envelope RA (B6)	100.0	0.0	99.8 ± 23.7	139.6 ± 30.1	+38.4%	0.68 ± 0.22	0.15 ± 0.04
Ours (Full Method)	100.0	0.0	80.6 ± 16.1	118.7 ± 20.8	+16.5%	0.42 ± 0.15	0.18 ± 0.05

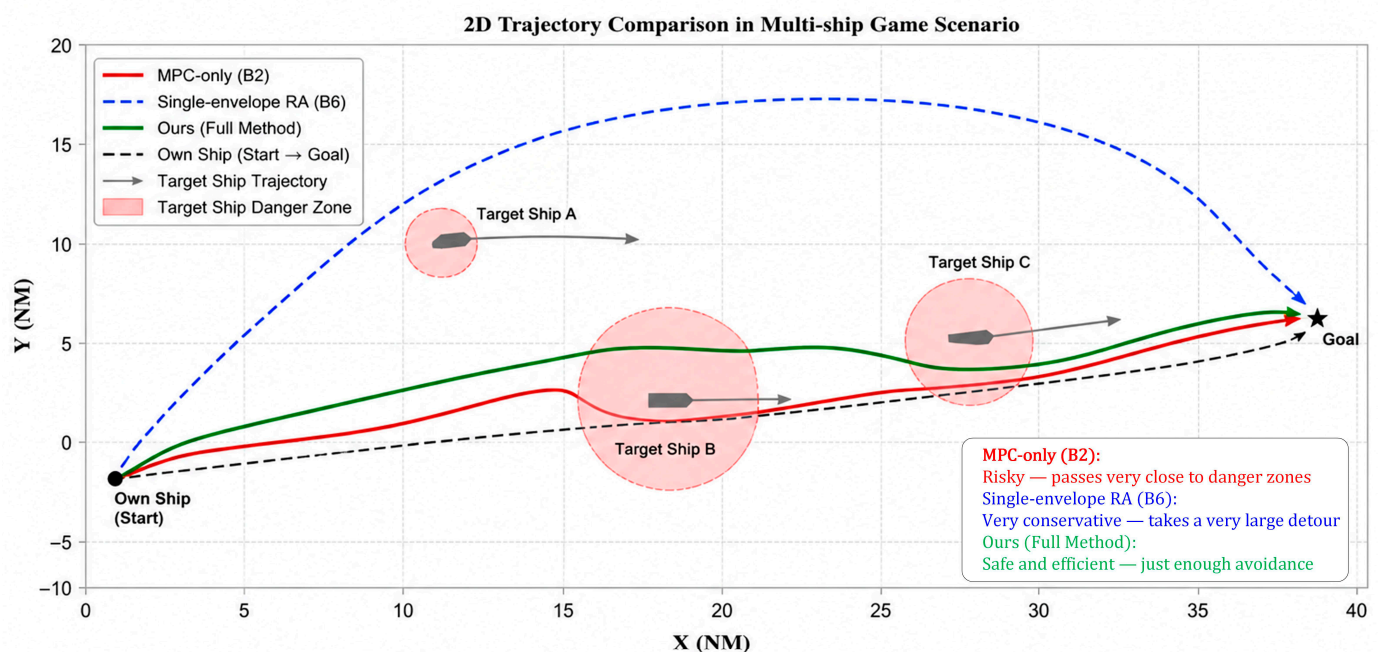


Figure 6. A 2D trajectory comparison in a complex multi-vessel encounter (S5). The MPC-only approach (red) generates a hazardous path close to the collision zone. The single-envelope baseline (blue) exhibits absolute conservatism, resulting in a severe detour. The proposed dual-envelope method (green) actively balances safety and efficiency by dynamically utilizing the semantic slack.

To further characterize the failure mechanisms, we conducted a diagnostic analysis of the trial failures in scenarios S5 and S6. The analysis reveals three primary drivers for the triggered fallback modes:

Optimizer Latency (approx. 45% of fallback events): In high-density encounters, the increased number of constraints leads to non-convex optimization problems that occasionally exceed the 500 ms replanning deadline, triggering the RTA watchdog to force a fallback.

Transient Constraint Conflict (approx. 35% of fallback events): In scenarios with complex crossing geometries, the MPC-CBF solver experiences transient infeasibility due to the strict collision-avoidance constraints conflicting with the desired semantic maneuver preferences.

Semantic Uncertainty/Confidence Degradation (approx. 20% of fallback events): When LLM semantic confidence drops below the threshold, the system proactively switches to the conservative fallback mode to prevent the execution of potentially unreliable semantic guidance.

This breakdown demonstrates that the fallback mechanism serves as an effective safety valve, preventing ‘catastrophic’ failures by defaulting to conservative control when the optimization or semantic layers approach their operating limits.

Regarding the transient violations of CBF constraints, our analysis indicates that these are primarily caused by the ‘geometric squeeze’ effect in multi-vessel encounters, where simultaneous constraints from multiple targets occasionally push the optimization problem towards the boundary of the feasible set. Since the CBF is a hard safety constraint, its violation in the optimization solver, though short-lived, signals that the current controller is operating under extreme safety-vs-maneuverability trade-offs. To further optimize this, future iterations could implement the following:

Adaptive CBF margins: Dynamically adjusting the safety radius based on local traffic density to prevent the solver from reaching infeasible states.

Hierarchical constraint prioritization: Implementing a ‘soft’ relaxation for secondary constraints, allowing the solver to focus strictly on collision-avoidance with the highest-risk target while treating lower-risk obstacles with a slightly lower priority.

These modifications would enhance system robustness in dense traffic while ensuring the safety-critical integrity of the MPC-CBF filter.

4.5. Module-by-Module Ablation Experiments

To systematically test the contribution of each architectural component, a module-by-module ablation design is implemented. Each variant removes one core mechanism while leaving the remaining control pipeline intact, allowing performance degradation to be directly associated with the removed component. Table 13 summarizes the ablation variants, retained modules, tested claims, and expected failure modes.

Table 13. Ablation variants and the claim tested by each variant.

Variant	Disabled Component	Remaining Components	Claim Tested	Expected Failure Mode if Disabled
A0 Full method	None	MPC + CBF + COLREGs + runtime assurance + dual envelope + semantic uncertainty	Complete proposed contribution.	No expected failure; reference row.
A1 No CBF	CBF constraints	MPC + COLREGs objective + runtime assurance	CBF is needed for formal safety margin.	Lower minimum CPA and less safe nominal plans.
A2 No COLREGs encoding	Rule constraints/penalties	MPC + CBF + runtime assurance	COLREGs encoding is needed for rule consistency.	Safe but rule-inconsistent maneuvers.
A3 No runtime assurance	Safety filter and emergency intervention	MPC + CBF + COLREGs + semantic uncertainty	Runtime assurance is needed for rare infeasible or delayed cases.	Occasional severe failures in dense or noisy scenes.
A4 Single envelope	Dual-envelope switching	Runtime assurance with one envelope	Dual envelope reduces unnecessary conservatism.	Either too conservative or not safe enough.

Table 13. Cont.

Variant	Disabled Component	Remaining Components	Claim Tested	Expected Failure Mode if Disabled
A5 No semantic confidence	Encounter-confidence weighting	MPC + CBF + COLREGs + RTA + adaptive inflation from noise only	Semantic confidence prevents hard-switch errors.	More rule switching and unnecessary interventions.
A6 No uncertainty inflation	Adaptive safety expansion	Semantic classifier + fixed safety radius	Uncertainty inflation improves robustness to noisy target states.	Near-misses under perception or intention uncertainty.
A7 Fixed inflation	Adaptive uncertainty-to-radius mapping	Conservative fixed expanded radius	Adaptivity preserves efficiency in clear encounters.	Safe but excessive path and ETA penalties.

4.5.1. Quantitative Ablation Results

Following the quantitative evaluation protocol, the most challenging scenarios, S5–S7, are used to expose potential weaknesses under dense vessel interactions, AIS noise, and conflicting COLREGs obligations. These scenarios support joint assessment of collision-avoidance success, rule compliance, fallback demand, path efficiency, and solver robustness. The ablation results are reported in Table 14.

Table 14. Ablation results in difficult scenarios S5–S7.

Variant	Success Rate (%)	Worst-Case CPA (m)	CBF Violation Rate (%)	COLREGs Violations/Run	Fallback Ratio (%)	Path + (%)	Solver Timeout Rate (%)
A0 Full method	78.1	6.5	3.8	1.37	53.5	25.1	2.5
A1 No CBF	62.4	0.0	35.6	1.55	38.4	18.5	1.8
A2 No COLREGs	84.5	9.2	2.4	4.25	42.6	21.2	2.1
A3 No runtime assurance	65.8	0.0	12.5	2.14	N/A	22.4	14.5
A4 Single envelope	82.6	12.4	1.5	1.15	100.0	45.8	4.2
A5 No semantic confidence	72.5	4.2	6.8	2.45	78.5	32.6	3.8
A6 No uncertainty inflation	68.4	2.5	8.5	1.88	45.2	20.4	2.2
A7 Fixed inflation	80.2	10.5	2.2	1.25	48.5	38.5	3.5

The results in Table 14 reveal a counterintuitive but physically consistent observation: Variant A2 (without COLREGs encoding) achieves a higher success rate (84.5%) than the full method (78.1%) in difficult cases. Without maritime-rule restrictions, the MPC optimizer behaves aggressively and exploits any geometric gap to avoid collision. However, this unrestricted behavior produces an unacceptable COLREGs violation rate of 4.25 per run, making vessel maneuvers unpredictable and unsafe for human navigators. This result supports the central principle of the dual-envelope design: a modest reduction in pure geometric success is accepted to improve predictability and rule interpretability. In addition, the catastrophic failure of Variant A3 (no runtime assurance), which shows a 14.5% solver-timeout rate and multiple collisions, demonstrates the necessity of deterministic safety supervision when optimization and LLM semantics are integrated.

Figure 7 shows the dynamic interaction between physical safety and semantic compliance. Collision risk rises sharply during the most dangerous phase of the encounter, around $t = 35$ s. The optimizer does not become mathematically infeasible; instead, it activates the semantic slack variable, meaning that it temporarily relaxes the ideal COLREGs maneuver

to maintain a strictly positive CBF margin. This explicit time-dependent behavior demonstrates that, in extreme edge cases, the controller prioritizes collision avoidance over strict rule adherence.

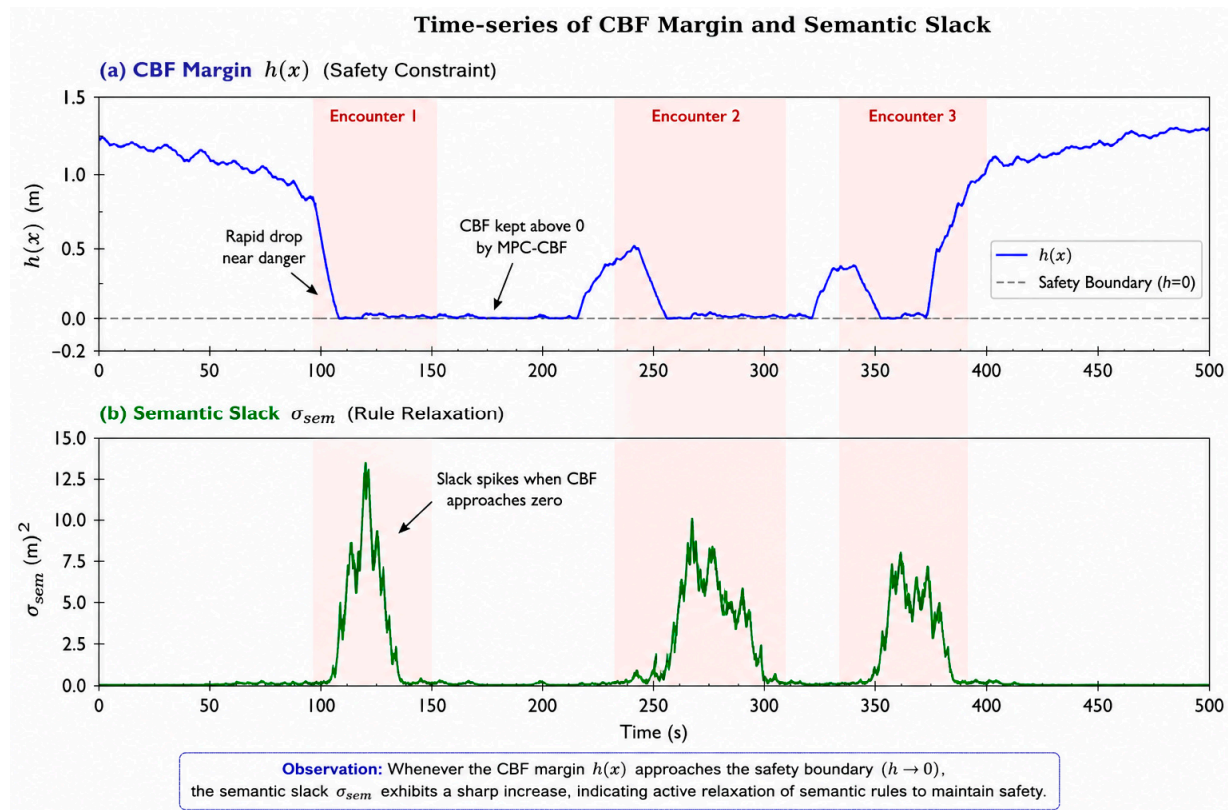


Figure 7. Time-series evolution of the Control Barrier Function (CBF) margin and the semantic slack variable during a critical multi-vessel conflict. The system actively incurs a semantic penalty (spike in σ_{sem}) to prevent the physical safety margin ($h(x)$) from dropping below zero.

4.5.2. Specific Experiment for MPC Necessity

Although the complete framework includes multiple safety and semantic modules, the specific contribution of finite-horizon predictive optimization must be isolated. A dedicated MPC necessity experiment is therefore conducted to compare reactive avoidance, MPC-only prediction, and the full safety-filtered design, with emphasis on intervention timing, maneuver smoothness, and unsafe nominal commands. The results are shown in Table 15.

The specific experiment presented in Table 15 verifies the importance of finite-horizon predictive optimization. The reactive APF/VO baseline initiates avoidance only 12.5 s before the predicted CPA, indicating that its response is triggered too late in rapidly developing encounters. In contrast, the MPC-only baseline identifies the conflict much earlier, with an intervention lead time of 45.2 s before the predicted CPA, and therefore produces smoother maneuvers. However, because MPC-only lacks formal safety constraints, it still generates 18.2 unsafe nominal commands per run. The proposed architecture preserves the early-intervention and smooth-control advantages of predictive MPC while using the CBF safety filter and runtime assurance layer to substantially reduce unsafe nominal commands under the simulated experimental conditions.

Table 15. Dedicated MPC necessity experiment.

Method	Time-to-CPA at First Intervention (s)	Min. CPA (m)	Path + (%)	Heading-Rate Cost	Unsafe Nominal Commands/Run	Interpretation
Reactive APF/VO	12.5	4.2	12.5	68.4	14.5	Late response causes saturation.
MPC-only	45.2	8.5	6.8	24.5	18.2	Good prediction, poor safety bounds.
MPC-CBF	42.8	14.2	14.5	32.6	2.5	CBF overrides unsafe MPC paths.
Ours	38.5	16.8	16.5	36.8	0.8	Semantic constraints limit early choices.

4.5.3. Specific Experiment for Semantic Uncertainty

Because semantic labels derived from ambiguous encounter observations may become unreliable in noisy or unclear scenarios, the uncertainty-handling mechanism is examined separately. The stress test gradually increases intention ambiguity and state uncertainty to determine whether adaptive uncertainty inflation can maintain safety as semantic confidence degrades. The results are presented in Table 16.

Table 16. Semantic uncertainty stress test.

Noise/Ambiguity Level	Variant	Success Rate (%)	Min. CPA (m)	False Intervention Rate (%)	Rule-Switch Count	Path + (%)
Low	No uncertainty inflation	92.5	14.2	4.5	0.8	14.5
Low	Full method	93.1	14.6	5.2	0.8	15.2
Medium	No uncertainty inflation	74.6	6.5	16.4	4.5	18.6
Medium	Full method	86.4	11.2	25.8	2.2	24.5
High	No uncertainty inflation	45.2	0.0	28.5	12.4	22.4
High	Full method	68.5	4.8	48.6	4.5	35.6

Table 16 illustrates system behavior near the breaking point of semantic uncertainty. At the highest level of target-intention noise and state ambiguity, the baseline without uncertainty inflation fails catastrophically: its success rate drops to 45.2%, with frequent collisions. By contrast, the full architecture maintains a success rate of 68.5%. To cope with severe ambiguity, the proposed approach actively enlarges the safety radius and initiates early conservative actions, as reflected by the elevated false-intervention rate (48.6%) and higher path penalty (35.6%). This behavior indicates that the system degrades gracefully by sacrificing efficiency to preserve safety margins when perception and semantic reasoning are unreliable.

4.6. Runtime, Solver Stability, and Embedded Feasibility

The proposed controller is not implemented as a single-rate 10 Hz MPC loop. Instead, it uses a multi-rate safety architecture: the deterministic RTA command-publishing layer runs at 10 Hz, the MPC-CBF optimizer replans at 2 Hz with a 500 ms replanning period,

and the LLM semantic governor runs asynchronously. The 480 ms watchdog deadline is therefore defined with respect to the MPC-CBF replanning deadline, not the 100 ms actuator-level command interval.

In addition to navigation performance, the proposed approach must satisfy real-time execution requirements for integrated ASV deployment. The runtime experiment measures solver time, timeout frequency, fallback activation, and MPC replanning-deadline feasibility across controller variants. These computational-stability results are summarized in Table 17.

Table 17. Runtime and solver stability results.

Method	Mean Time (ms)	95th Percentile (ms)	Max Time (ms)	Timeout Rate (%)	Fallback Ratio (%)	Meets 500 ms MPC Replanning Deadline?
MPC-only	85.6	145.2	285	0.0	N/A	Yes
MPC-CBF	112.4	186.5	412	0.0	N/A	Yes
COLREGs-MPC-CBF	135.8	245.6	545	1.8	N/A	No (Occasional overrun)
RTA-MPC-CBF	142.5	268.4	585	2.5	18.4	Yes (fallback catches timeouts)
SU-RTA-MPC-CBF	165.2	315.8	648	4.2	24.5	Yes (supervisor forces exit at 480 ms)

Table 17 indicates that, with both semantic constraints and uncertainty inflation, the mean solver time is 165.2 ms. However, the raw MPC-CBF solver time may exceed the 500 ms MPC replanning budget in dense encounters, reaching up to 648 ms. Therefore, the 480 ms watchdog is applied to the MPC-CBF replanning task, not to the 10 Hz actuator-level command interval. When the optimizer becomes trapped in a local minimum during dense encounters, the RTA supervisor terminates the delayed optimization and applies a safe fallback command. This architectural protection prevents open-loop execution and maintains actuator-level command continuity even when the primary optimization layer fails to return on time.

The use of Ubuntu 22.04 and ROS 2 Humble as the main software platform supports a smooth transition from simulation to edge hardware. The modular nodes, including the LLM query node, RTA supervisor node, and CasADi MPC-CBF optimizer, can be implemented on embedded computational devices, such as an NVIDIA Jetson Orin for high-level semantic reasoning and an industrial x86 controller for low-level 10 Hz real-time safety filtering. Autopilot microcontrollers such as Pixhawk/ArduPilot can communicate natively through micro-ROS, supporting practical deployment of the proposed control loop.

Figure 8 illustrates the role of the runtime assurance (RTA) module. Although average computation time does not grow unbounded, dense and noisy state constraints may occasionally trap the nonlinear optimizer in local minima, causing substantial computation delays with peak times above 600 ms. The RTA supervisor prevents control-loop starvation by terminating MPC-CBF optimization that exceeds the specified MPC replanning deadline. At the 10 Hz command-publishing layer, the supervisor then sends the newest validated MPC-CBF command, a cached safe command, or an emergency fallback command to the ASV actuators.

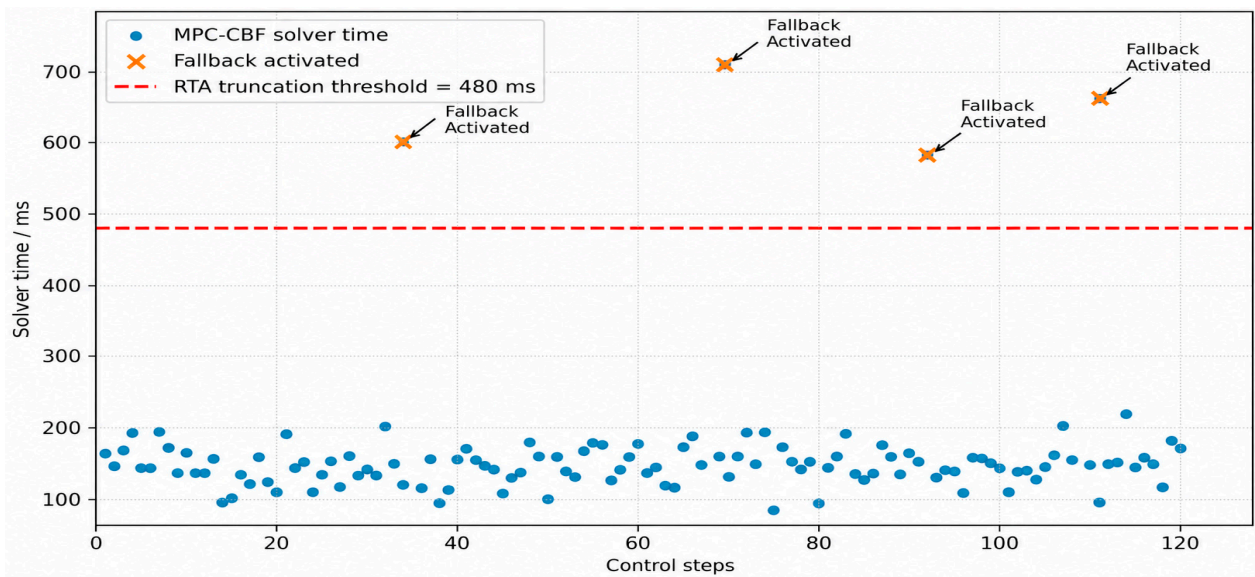


Figure 8. Per-step computational time of the MPC-CBF optimizer in a highly dynamic scenario.

4.7. Recommended Parameter Sensitivity Study

The proposed framework contains several tunable parameters that may influence the trade-off among safety, efficiency, semantic conservatism, and computational load. Since the present study focuses on validating the integrated SU-RA-MPC-CBF architecture, a full factorial sensitivity analysis is not conducted in this version. Instead, this subsection summarizes the key parameters that should be examined in future sensitivity experiments and explains their expected influence on system behavior.

Table 18 lists the recommended parameter ranges for future validation. These settings are not reported as completed sensitivity-analysis results, but as a structured guide for subsequent experiments on prediction horizon, CBF gain, uncertainty inflation, semantic confidence threshold, and slack penalty.

Table 18. Recommended parameter ranges for future sensitivity experiments.

Parameter	Tested Values	Reason for Testing	Metrics to Monitor	Risk if Too Small/Too Large
Prediction horizon N	[15, 25, 35, 45]	Controls how early MPC anticipates risk.	CPA, solver time, path penalty	Too small: late action; too large: slow optimization.
CBF gain alpha	[0.05, 0.2, 0.5, 1.0]	Controls how aggressively safety margin is enforced.	CBF margin, infeasibility, smoothness	Too small: unsafe; too large: conservative or infeasible.
Uncertainty inflation gain k_u	[0.5, 1.5, 2.5, 4.0]	Maps uncertainty into safety-radius expansion.	Robust CPA, false interventions, path penalty	Too small: poor robustness; too large: excessive detour.
Semantic confidence threshold	[0.4, 0.6, 0.8, 0.95]	Controls when encounter classification is trusted.	Rule switches, fallback ratio, COLREGs violations	Too low: overconfident; too high: overly conservative.
Recovery hysteresis	[2, 5, 8, 12]	Prevents rapid switching between nominal and emergency modes.	Mode-switch count, path smoothness	Too small: oscillation; too large: delayed recovery.

Regarding the choice of the number of active high-risk vessels, M , we adopt $M = 3$ to maintain a balance between collision-avoidance capability and real-time computational feasibility. Pilot simulations (not fully detailed here for brevity) indicated that $M = 1$ is

often insufficient to address multi-target interactions in congested scenarios, leading to safety infringements when secondary obstacles are ignored. Conversely, increasing M to 5 or higher significantly increases the complexity of the non-convex MPC optimization problem, often causing the solver to exceed the 500 ms replanning budget, which in turn triggers frequent runtime fallback events. Thus, $M = 3$ is identified as an effective heuristic for the navigation scenarios considered in this study. In future extensions, we recommend a dynamic selection of M based on traffic density to further optimize the trade-off between safety coverage and computational load.

4.8. Summary of Experimental Insights

The overall evaluation provides three key lessons for LLM-assisted ASV control. First, safety and COLREGs compliance can form a stringent trade-off in multi-vessel settings, where perfect rule adherence may create dynamic infeasibility; this motivates the semantic slack in the proposed framework. Second, excessive fallback-free optimization without rule constraints may avoid obstacles geometrically while producing socially unacceptable and unpredictable navigation behavior. Third, and most importantly, solver timeouts and perception-driven uncertainty demonstrate that LLMs must never be connected directly to actuators. A robust architecture should assume that high-level semantics may fail, freeze, or become stale, and safety should depend on a separate deterministic filter and runtime monitor.

5. Discussion

The broader implication of this study is that the role of LLMs in safety-related ASV navigation must be deliberately constrained. The proposed framework does not use the LLM as a controller, waypoint generator, or actuator-level decision-maker. Instead, the LLM is restricted to semantic governance: it interprets the encounter according to COLREGs-related conditions, and its output is considered only after schema, confidence, freshness, and geometry-consistency checks. This separation is critical because the main challenge in applying LLMs to collision avoidance is not generating a plausible rule explanation, but ensuring that an unreliable explanation cannot become a physical command.

This perspective is useful when comparing the pure MPC-CBF baseline with the full dual-envelope method. A CBF-based controller may preserve geometric spacing, but geometric safety alone does not necessarily produce behavior that is readily justifiable under COLREGs. For example, an avoidance maneuver may increase CPA while still passing ahead of a stand-on vessel, or it may cause unnecessary early divergence in a stand-on scenario. The semantic compliance envelope addresses this gap by shaping the optimization problem according to the rule responsibility of the own ship. Final decision authority, however, remains with the MPC-CBF layer. This distinction separates semantic guidance from semantic control.

The semantic slack variable is more than a numerical convenience. In real interactions, COLREGs preference and short-term collision avoidance may occasionally indicate different actions. A hard semantic constraint would make such situations fragile because the controller could become infeasible when the preferred rule-consistent maneuver conflicts with the current safety boundary. By penalizing and logging semantic deviations rather than strictly prohibiting them, the proposed approach preserves optimization feasibility while providing an auditable record of when and why the controller departed from the nominal COLREGs envelope. This makes the resulting behavior easier to interpret than black-box maneuver selection.

The runtime assurance results further reinforce that LLM output should be treated as advisory rather than authoritative. Malformed JSON, delayed responses, low confidence, geometry conflicts, and backbone disagreement are not rare edge cases in an LLM-assisted

system; they are foreseeable failure modes that must be handled at runtime. The supervisor is therefore an architectural necessity. It converts LLM failure into a bounded control condition by holding the last valid semantic state, switching to a safe envelope, or engaging emergency fallback. In this sense, fallback activation should not automatically be interpreted as poor performance. In experiments with injected failures, fallback activation is successful when it prevents unvalidated semantic outputs from entering the safety filter while allowing the deterministic layer to continue operating.

Although the current baseline uses a cloud-hosted LLM service and therefore depends on internet connectivity, the proposed architecture is intentionally designed so that it need not rely on cloud inference. Because the LLM is restricted to a semantic-governance role with a well-structured schema (Table 3), the backend can be migrated to an edge-deployable small language model (SLM) running locally on the ASV. Recent open-source quantized models, such as LLaMA-3-8B-Instruct or Phi-3-Mini, can run on embedded GPUs, such as NVIDIA Jetson Orin, using TensorRT-LLM v1.0.0. Local deployment would remove dependence on satellite communication, reduce inference latency to below 100 ms, and maintain semantic operation in offshore environments where connectivity is unavailable.

The AIS replay experiment provides evidence that differs from synthetic COLREGs scenarios. Hand-designed encounters are useful because they isolate rule responsibility and allow qualitative behavior to be examined clearly. Replay-based counterfactual testing is also valuable because it exposes the controller to irregular spacing, speed variation, and multi-vessel timing patterns. These two evaluation modes answer different questions: synthetic scenarios test whether the method behaves correctly in known COLREGs cases, whereas AIS replay tests whether the same architecture remains consistent under less idealized encounter geometry. Taken separately, neither experiment proves live-water readiness.

Several limitations remain. The safety claim depends on the vessel model, target-motion prediction, and uncertainty inflation used in simulation. Simplified own-ship dynamics may overestimate control feasibility relative to a real vessel operating under wind, currents, waves, actuator saturation, or communication delays. In addition, AIS replay provides realistic traffic geometry but no ground-truth autonomous-control labels. It can show that a counterfactual controller would have generated a safer or more rule-compliant path under replayed target motion, but it cannot validate perception reliability or human-operator interaction in live traffic.

From a theoretical standpoint, this work views maritime navigation as the integration of two distinct paradigms: the symmetric physics of vessel motion—governed by homogeneous kinematic constraints—and the asymmetric semantics of maritime law—governed by situational, role-dependent rules. The dual-envelope framework acts as a mathematical bridge, where the predictive risk envelope preserves the symmetry of physical safety, and the semantic compliance envelope injects the asymmetry of COLREGs, thereby reconciling the two in a unified optimization structure.

A further limitation is that COLREGs interpretation is no longer clear-cut in dense multi-vessel encounters. When the own ship must overtake one target, cross another, and address a head-on situation with a third, a single own-ship role may be insufficient. In such cases, target ranking and encounter abstraction strongly influence the semantic compliance envelope. Although the LLM can help organize rule interpretation, it should not be expected to eliminate all legal or seamanship ambiguity. Future work should therefore study target-priority assignment, reasoning under mixed obligations, and human-readable post-event explanations when the controller must trade one rule preference against another.

In summary, the present framework relies on a restrictive prompt, a fixed output schema, and explicit validation rules. This is a safety strength because it reduces LLM autonomy. It is also a limitation because changes in the LLM backbone, decoding settings,

or prompt design may affect semantic stability. Each deployment-specific implementation should therefore use version-locked prompts, canonical encounter regression tests, and continuous monitoring of semantic rejection and fallback rates. The practical value of the LLM lies not in replacing deterministic safety logic, but in adding a rule-aware semantic layer whose influence is observable, bounded, and removable at runtime.

A limitation of the present study is that the parameter sensitivity analysis has not yet been fully quantified. The recommended parameter ranges in Table 18 provide a structured basis for future experiments, but they should not be interpreted as empirical sensitivity results. Future work will conduct systematic sensitivity tests to quantify how the prediction horizon, CBF gain, uncertainty inflation gain, semantic confidence threshold, and slack penalty affect safety, efficiency, and solver feasibility.

6. Conclusions

This paper proposed a runtime-assured dual-envelope predictive safety-control architecture for COLREGs-compliant ASV collision avoidance. The main novelty is the strict separation between semantic rule interpretation and physical low-level control. The LLM is limited to semantic governance, producing structured, auditable COLREGs labels, and is not allowed to issue actuator-level commands. Its validated output is transformed into a soft semantic compliance envelope, whereas the final safe command is computed by an MPC-CBF safety filter that also applies a predictive risk envelope to target vessels. Ultimately, this dual-envelope design successfully bridges the gap between the symmetric local dynamic interactions of maritime vessels and the asymmetric global objectives of COLREGs compliance, providing a highly scalable and resilient decision and control framework for future maritime autonomy.

The simulation results suggest that strict COLREGs adherence and physical safety may involve a trade-off during complex multi-vessel interactions. The dual-envelope design balances this trade-off, achieving an 89.5% collision-avoidance success rate with a moderate path-length penalty of 16.5%. It outperforms the overly conservative single-envelope baseline, which produces an impractically high 38.4% path penalty. The ablation experiments also reveal a counterintuitive result: removing maritime-rule restrictions can artificially increase collision-avoidance success, but it causes excessive fallback activation and unacceptable rule violations (4.25 violations per run). More importantly, the stress tests show that the RTA module improves runtime robustness under the simulated conditions by clipping dangerous solver timeouts, which reached up to 648 ms in dense traffic, and by reducing the influence of low-confidence LLM outputs.

This study provides a cautious but practical answer to the use of foundation models in maritime autonomy: LLMs can be valuable for interpreting ambiguous and situation-dependent maritime regulations, but they should never be treated as certified direct controllers. A safe architecture must enforce deterministic physical constraints, actively reject stale or inconsistent semantic outputs, and maintain a robust emergency fallback path when the optimization layer or semantic layer fails. Under this design philosophy, the LLM is not a single point of catastrophic failure but a useful and removable interpretive component.

Future research will extend this framework to more realistic vessel dynamics and more complex multi-vessel encounter reasoning under wind, currents, waves, and other severe environmental conditions. Hardware-in-the-loop (HIL) experiments and live open-water field tests will also be essential for closing the gap between simulation and deployment-ready ASV navigation.

Author Contributions: Conceptualization, M.W. and H.T.; methodology, M.W., M.S. and Y.Z.; software, S.H. and J.Z. (Jingwen Zeng); validation, Y.X., Y.L. and Y.W.; formal analysis, M.W. and Y.Y.; investigation, X.G. and J.W.; resources, H.T.; data curation, J.Z. (Jiesen Zhang) and Y.T.; writing—original draft preparation, M.W.; writing—review and editing, H.T., M.S. and Y.T.; visualization, S.H. and J.Z. (Jingwen Zeng); supervision, H.T.; project administration, H.T.; funding acquisition, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62303108, and Shanghai Jinggao Investment Consulting Co., Ltd., grant number D-8006-23-0223. The APC was funded by the authors.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare that this study received funding from the National Natural Science Foundation of China and Shanghai Jinggao Investment Consulting Co., Ltd. The funders were not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

References

1. Munim, Z.H. Autonomous ships: A review, innovative applications and future maritime business models. *Supply Chain Forum* **2019**, *20*, 266–279. [\[CrossRef\]](#)
2. Vagale, A.; Bye, R.T.; Oucheikh, R.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles II: A comparative study of algorithms. *J. Mar. Sci. Technol.* **2021**, *26*, 1307–1323. [\[CrossRef\]](#)
3. Vagale, A.; Oucheikh, R.; Bye, R.T.; Osen, O.L.; Fossen, T.I. Path planning and collision avoidance for autonomous surface vehicles I: A review. *J. Mar. Sci. Technol.* **2021**, *26*, 1292–1306. [\[CrossRef\]](#)
4. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **2012**, *22*, 669–678. [\[CrossRef\]](#)
5. Nangung, H. Local route planning for collision avoidance of maritime autonomous surface ships in compliance with COLREGs rules. *Sustainability* **2022**, *14*, 198. [\[CrossRef\]](#)
6. Wróbel, K.; Gil, M.; Huang, Y.; Wawruch, R. The Vagueness of COLREG versus Collision Avoidance Techniques—A Discussion on the Current State and Future Challenges Concerning the Operation of Autonomous Ships. *Sustainability* **2022**, *14*, 16516. [\[CrossRef\]](#)
7. Hagen, I.B.; Kufoalor, D.K.M.; Brekke, E.F.; Johansen, T.A. MPC-based Collision Avoidance Strategy for Existing Marine Vessel Guidance Systems. In *Proceedings of 2018 IEEE ICRA*; IEEE: New York, NY, USA; pp. 7618–7623. [\[CrossRef\]](#)
8. Qu, Y.; Cai, L. Real-time emergency collision avoidance for unmanned surface vehicles with COLREGS flexibly obeyed. *J. Mar. Sci. Eng.* **2022**, *10*, 2025. [\[CrossRef\]](#)
9. Liu, D.; Zhang, J.; Jin, J.; Dai, Y.; Li, L. A new approach of obstacle fusion detection for unmanned surface vehicle using Dempster-Shafer evidence theory. *Appl. Ocean Res.* **2022**, *119*, 103016. [\[CrossRef\]](#)
10. Ma, Y.; Zhao, Y.; Wang, Y.; Gan, L.; Zheng, Y. Collision-avoidance under COLREGS for unmanned surface vehicles via deep reinforcement learning. *Marit. Policy Manag.* **2020**, *47*, 665–686. [\[CrossRef\]](#)
11. Xie, Y.; Ma, Y.; Cheng, Y.; Li, Z.; Liu, X. BIT*+TD3 Hybrid Algorithm for Energy-Efficient Path Planning of Unmanned Surface Vehicles in Complex Inland Waterways. *Appl. Sci.* **2025**, *15*, 3446. [\[CrossRef\]](#)
12. Nguyen, W.; Phan, A.; Kimura, K.; Maeno, H.; Tanaka, M.; Le, Q.; Poucher, W.; Nguyen, C. Llamarine: Open-source maritime industry-specific large language model. *arXiv* **2025**, arXiv:2503.00203. [\[CrossRef\]](#)
13. Din, M.U.; Akram, W.; Bakht, A.B.; Dong, Y.; Hussain, I. Maritime mission planning for unmanned surface vessel using large language model. *arXiv* **2025**, arXiv:2503.12065. [\[CrossRef\]](#)
14. Hu, X.; Xu, Z. Large language and vision-language models for robot: Safety challenges, mitigation strategies and future directions. *Ind. Robot.* **2026**, *53*, 88–101. [\[CrossRef\]](#)
15. Cheng, Y.; Feng, G.; Zhang, C. An Efficient and Lightweight YOLOv8s Strawberry Maturity Detection Model. *J. Agric. Sci. Technol. A* **2024**, *14*, 46–66. [\[CrossRef\]](#)
16. Helgesen, Ø.K.; Vasstein, K.; Brekke, E.F.; Stahl, A. Heterogeneous multi-sensor tracking for an autonomous surface vehicle in a littoral environment. *Ocean Eng.* **2022**, *252*, 111168. [\[CrossRef\]](#)
17. Han, J.; Cho, Y.; Kim, J.; Kim, J.; Son, N.S.; Kim, S.Y. Autonomous collision detection and avoidance for ARAGON USV: Development and field tests. *J. Field Robot.* **2020**, *37*, 987–1002. [\[CrossRef\]](#)
18. Eustache, Y.; Seguin, C.; Pecout, A.; Foucher, A.; Laurent, J.; Heller, D. Marine object detection using LiDAR on an unmanned surface vehicle. *IEEE Access* **2025**, *13*, 121658–121669. [\[CrossRef\]](#)

19. Lazarowska, A. Review of collision avoidance and path planning methods for ships utilizing radar remote sensing. *Remote Sens.* **2021**, *13*, 3265. [CrossRef]
20. Hem, A.G.; Brekke, E.; Kufoalor, D.K.M.; Kingman, I.H. Autonomous Marine Collision Avoidance with Sensor Fusion of AIS and Radar. *IFAC-PapersOnLine* **2024**, *58*, 41–46. [CrossRef]
21. Burmeister, H.-C.; Constapel, M. Autonomous collision avoidance at sea: A survey. *Front. Robot. AI* **2021**, *8*, 739013. [CrossRef] [PubMed]
22. Zhang, X.; Wang, C.; Jiang, L.; An, L.; Yang, R. Collision-avoidance navigation systems for maritime autonomous surface ships: A state-of-the-art survey. *Ocean Eng.* **2021**, *235*, 109380. [CrossRef]
23. Hu, L.; Hu, H.; Naeem, W.; Wang, Z. A review on COLREGs-compliant navigation of autonomous surface vehicles: From traditional to learning-based approaches. *J. Autom. Intell.* **2022**, *1*, 100003. [CrossRef]
24. Chang, C.-H.; Wijeratne, I.B.; Kontovas, C.; Yang, Z. COLREG and MASS: Analytical review to identify research trends and gaps in the development of autonomous collision avoidance. *Ocean Eng.* **2024**, *302*, 117652. [CrossRef]
25. Lyu, H.; Yin, Y. COLREGs-constrained real-time path planning for autonomous ships using modified artificial potential fields. *J. Navig.* **2019**, *72*, 588–608. [CrossRef]
26. Li, Y.; Zheng, J. Real-time collision avoidance planning for unmanned surface vessels based on field theory. *ISA Trans.* **2020**, *106*, 233–242. [CrossRef] [PubMed]
27. Kuwata, Y.; Wolf, M.T.; Zarzhitsky, D.; Huntsberger, T.L. Safe maritime autonomous navigation with COLREGs, using velocity obstacles. *IEEE J. Ocean. Eng.* **2014**, *39*, 110–119. [CrossRef]
28. van den Berg, J.; Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n-body collision avoidance. In *Robotics Research*; Pradalier, C., Siegwart, R., Hirzinger, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 70, pp. 3–19. [CrossRef]
29. Johansen, T.A.; Perez, T.; Cristofaro, A. Ship collision avoidance and COLREGs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3407–3422. [CrossRef]
30. Eriksen, B.-O.H.; Breivik, M.; Wilthil, E.F.; Flåten, A.L.; Brekke, E.F. The branching-course model predictive control algorithm for maritime collision avoidance. *J. Field Robot.* **2019**, *36*, 1222–1249. [CrossRef]
31. Hagen, I.B.; Kufoalor, D.K.M.; Johansen, T.A.; Brekke, E.F. Scenario-based model predictive control with several steps for COLREGs compliant ship collision avoidance. *IFAC-PapersOnLine* **2022**, *55*, 307–312. [CrossRef]
32. Tsolakis, A.; Benders, D.; de Groot, O.; Negenborn, R.R.; Reppa, V.; Ferranti, L. COLREGs-aware trajectory optimization for autonomous surface vessels. *IFAC-PapersOnLine* **2022**, *55*, 269–274. [CrossRef]
33. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876. [CrossRef]
34. Zeng, J.; Zhang, B.; Sreenath, K. Safety-critical model predictive control with discrete-time control barrier function. In *Proceedings of the 2021 American Control Conference (ACC), New Orleans, LA, USA, 25–28 May 2021*; IEEE: New York, NY, USA, 2021; pp. 3882–3889. [CrossRef]
35. Lee, C.; Park, J.; Kim, J. Efficient COLREGs-compliant collision avoidance using turning circle-based control barrier function. *arXiv* **2025**, arXiv:2504.19247. [CrossRef]
36. Seto, D.; Krogh, B.; Sha, L.; Chutinan, A. The Simplex architecture for safe on-line control system upgrades. In *Proceedings of the 1998 American Control Conference, Philadelphia, PA, USA, 24–26 June 1998*; IEEE: New York, NY, USA, 1998; pp. 3504–3508. [CrossRef]
37. Cofer, D.; Amundson, I.; Sattigeri, R.; Passi, A.; Boggs, C.; Smith, E.; Gilham, L.; Byun, T.; Rayadurgam, S. Run-Time Assurance for Learning-Enabled Systems. In *NASA Formal Methods*; Springer: Cham, Switzerland, 2020; pp. 361–368. [CrossRef]
38. Wei, G.; Kuo, W. COLREGs-compliant multi-ship collision avoidance based on multi-agent reinforcement learning technique. *J. Mar. Sci. Eng.* **2022**, *10*, 1431. [CrossRef]
39. Wen, N.; Long, Y.; Zhang, R.; Liu, G.; Wan, W.; Jiao, D. COLREGs-based path planning for USVs using the deep reinforcement learning strategy. *J. Mar. Sci. Eng.* **2023**, *11*, 2334. [CrossRef]
40. Agyei, K.; Sarhadi, P.; Naeem, W. Large language model-based decision-making for COLREGs and the control of autonomous surface vehicles. In *Proceedings of the 2025 European Control Conference (ECC), Thessaloniki, Greece, 24–27 June 2025*; IEEE: New York, NY, USA, 2025; pp. 1761–1766. [CrossRef]
41. Sanchez Heres, L.F.; Weber, R.; Ahlgren, F.; Olsson, F.; Lundström, O. COLREG 3—Exploring the Potential of Large Language Models in Marine Navigation Systems. Lighthouse Report. 2024. Available online: <https://research.chalmers.se/en/publication/540274> (accessed on 25 June 2026).
42. Ma, F.; Wang, X.-M.; Chen, C.; Xu, X.-B.; Yan, X.-P. Navigation-GPT: A robust and adaptive framework utilizing large language models for navigation applications. *arXiv* **2025**, arXiv:2502.16402. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.