

Article

Evolving Collective Intelligence for Unmanned Marine Vehicle Swarms: A Federated Meta-Learning Framework for Cross-Fleet Planning and Control

Yuhan Ye ¹, Hongjun Tian ^{1,*} , Yijie Yin ¹, Yuhan Zhou ², Yang Xiong ¹, Zi Wang ¹, Yaojiang Liu ¹, Zinan Nie ³, Zitong Zhang ⁴, Yichen Wang ¹ and Jingyu Sun ⁵

¹ Engineering College, Shanghai Ocean University, Shanghai 201306, China; 19852242618@163.com (Y.Y.); silence18177@163.com (Y.Y.); 13377022951@163.com (Y.X.); eason20070329@163.com (Y.W.)

² School of Merchant Marine, Shanghai Maritime University, Shanghai 201306, China; zhouyih2004@163.com

³ College of Marine Science and Ecological Environment, Shanghai Ocean University, Shanghai 201306, China; 19968061316@163.com

⁴ College of Fisheries and Life Sciences, Shanghai Ocean University, Shanghai 201306, China; jennyzitong@outlook.com

⁵ Aien College, Shanghai Ocean University, Shanghai 201306, China; jingyus29@163.com

* Correspondence: tianhongjun86@163.com

Abstract

The development of robust autonomous maritime systems is fundamentally constrained by the “data silo” problem, where valuable operational data from disparate fleets remain isolated due to privacy concerns, severely limiting the scalability of general-purpose navigation intelligence. To address this barrier, we propose a novel Federated Meta-Transfer Learning (FMTL) framework that enables collaborative evolution of unmanned surface vehicle (USV) swarms while preserving data privacy. Our hierarchical approach orchestrates three synergistic stages: (1) transfer learning pre-trains a universal “Sea-Sense” foundation model on large-scale maritime data to establish fundamental navigation priors; (2) federated learning enables decentralized fleets to collaboratively refine this model through encrypted gradient aggregation, forming a distributed cognitive network; (3) meta-learning allows for rapid personalization to individual vessel dynamics with minimal adaptation trials. Comprehensive simulations across heterogeneous fleet distributions demonstrate that our federated model achieves a 95.4% average success rate across diverse maritime scenarios, significantly outperforming isolated specialist models (63.9–73.1%), while enabling zero-shot performance of 78.5% and few-shot adaptation within 8–12 episodes on unseen tasks. This work establishes a scalable, privacy-preserving paradigm for collective maritime intelligence through swarm-based learning.

Keywords: federated learning; meta-learning; transfer learning; unmanned surface vehicle; autonomous navigation; swarm intelligence; data privacy; collision avoidance; deep reinforcement learning; maritime cognitive network



Academic Editor: Weicheng Cui

Received: 21 November 2025

Revised: 15 December 2025

Accepted: 18 December 2025

Published: 31 December 2025

Copyright: © 2025 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

1.1. The Grand Challenge: The Data Silo Dilemma in Maritime Autonomy

The pursuit of fully autonomous Unmanned Surface Vehicles (USVs) promises to revolutionize maritime operations, from sustainable ocean exploration and climate monitoring to secure global logistics [1,2]. However, despite significant advances in AI and robotics, the transition from controlled simulations to robust, real-world deployment remains fraught

with challenges. We argue that the most profound and systemic barrier is not algorithmic insufficiency, but a fundamental structural problem: the “data silo” dilemma.

In the real world, valuable navigation data—the lifeblood of learning-based AI—is fragmented across countless isolated “islands.” A commercial shipping company in Asia possesses vast datasets from navigating congested straits; a European oceanographic institute holds unique data from polar expeditions; a North American port authority has rich data on managing dense harbor traffic. Each dataset is a treasure trove of experience, yet it is fiercely protected due to commercial confidentiality, national security concerns, or regulatory constraints. Consequently, the development of maritime AI is trapped in a paradox: the very data needed to create truly general-purpose intelligence are inaccessible. Current approaches, which train models on limited, homogeneous datasets, inevitably produce “brittle specialists”—agents that perform well in their training environment but fail catastrophically when faced with the vast heterogeneity of the world’s oceans [3–5]. This “train once, fail everywhere” reality represents a critical bottleneck, rendering the dream of scalable, universally reliable maritime autonomy unattainable under the current paradigm.

1.2. A New Paradigm: Towards a Global Maritime Cognitive Network

To shatter these data silos, we propose a paradigm shift from developing isolated intelligences to cultivating a collaborative, decentralized cognitive ecosystem. We envision a Global Maritime Cognitive Network, where fleets and vessels worldwide can contribute to and benefit from a shared, continuously evolving navigation intelligence without ever exposing their private data. This vision reframes the challenge: instead of asking “How do we build a single smart USV?”, we ask, “How do we engineer a system that allows an entire global fleet to learn and evolve together as a single, distributed mind?”

This paper introduces the first concrete realization of this paradigm: a hierarchical Federated Meta-Transfer Learning (FMTL) framework. Our approach is not an incremental improvement but a fundamental rethinking of how maritime AI is developed, deployed, and sustained. It is designed from first principles to address three core pillars of this new ecosystem: privacy, collaboration, and personalization.

1.3. The FMTL Framework: A Three-Stage Cognitive Development Architecture

Our FMTL framework orchestrates a synergistic, three-stage learning process that mirrors a sophisticated cognitive development path, taking an agent from a novice to a globally informed, locally adapted expert:

Stage 1: Foundational Understanding via Transfer Learning. The journey begins by pre-training a universal “Sea-Sense” foundation model on massive, publicly available (or simulated) maritime datasets. This endows the model with a robust understanding of fundamental physics, COLREGs-based navigation etiquette, and general vessel dynamics, forming a powerful, universal prior.

Stage 2: Collaborative Evolution via Federated Learning. This is the core of our collaborative ecosystem. The pre-trained foundation model is distributed to multiple, geographically and operationally distinct fleets. Each fleet uses its private, local data to fine-tune the model. Critically, only the anonymized and encrypted model updates (gradients) are sent to a secure aggregation server—the raw data never leave the owner’s control. The server intelligently aggregates these updates to produce a new, more capable global model, which is then redistributed. This federated cycle allows the collective intelligence to learn from a diversity of experiences—ice navigation, swarm interactions, littoral operations—far beyond what any single entity could amass, all while guaranteeing data privacy.

Stage 3: Rapid Personalization via Meta-Learning. The final stage addresses deployment. When a new USV is commissioned, it is initialized not with a blank slate, but with

the powerful, globally informed federated model. This model then serves as an exceptional starting point for a meta-learning process. Through just a handful of in situ trials (few-shot adaptation), the agent rapidly fine-tunes the model to its specific vessel kinematics, sensor configurations, and immediate operational context, achieving peak performance with minimal data and time.

1.4. Contributions and Outline

To explicitly distinguish our original contributions from established methodologies incorporated in this work, we summarize our specific innovations as follows:

A Novel FMTL Protocol (Original Framework): While Federated Learning (FL) and Meta-Learning are established fields, this work proposes the first specific integration protocol for maritime systems. Our key novelty is the hierarchical pipeline where a federated global model is utilized not as a final inference engine, but as a mathematically optimal meta-initialization for local few-shot adaptation, effectively solving the “cold-start” problem for new vessels.

Graph-Gated Transformer Architecture (Original Algorithm): We introduce a novel Graph-Gated Transformer (GGT) mechanism within the PPO agent. Distinct from standard “black-box” Transformers or GNNs, our GGT constructs a dynamic “Tactical Relational Graph” based on maritime rules (COLREGs) and injects it as a hard attention mask. This forces the neural network to prioritize physically relevant vessels over background noise, a significant departure from standard attention mechanisms.

Hybrid Safety Integration (System Design): We uniquely integrate these original learning components with valid standard modules—specifically the Theta algorithm* for global path planning and Control Barrier Functions (CBFs) for safety guarantees. The contribution lies in the architectural design that fuses these established deterministic methods with our novel stochastic learning agent to ensure “Sim-to-Real” viability.

2. Related Work

Our work introduces a novel, hierarchical learning paradigm to address the full lifecycle of maritime autonomous intelligence, from foundational knowledge acquisition to collaborative evolution and rapid personalization. This section reviews the three core learning paradigms we synthesize—Federated Learning, Transfer Learning, and Meta-Learning—contextualizing their individual strengths and limitations, thereby motivating the necessity of our integrated Federated Meta-Transfer Learning (FMTL) framework.

2.1. Federated Learning: A Key to Unlocking Collaborative Intelligence Under Privacy Constraints

The core challenge of the “data silo” dilemma is fundamentally one of privacy and data governance. Federated Learning (FL) has emerged as a revolutionary paradigm to address this very issue [6]. The foundational principle of FL, particularly horizontal FL as typified by the FedAvg algorithm, is to “move the model, not the data” [7]. In this framework, multiple clients (e.g., different corporate fleets) collaboratively train a shared global model by only exchanging anonymized and often encrypted parameter updates (gradients), while their raw, sensitive data remain securely on premise [8].

This privacy-preserving nature has made FL a key enabling technology in data-sensitive domains such as healthcare [9], where models are trained on patient data from multiple hospitals without violating confidentiality, and finance [10], for fraud detection across different banks. In the domain of autonomous systems, FL has seen initial exploration for tasks like collaborative perception in connected vehicles and trajectory prediction. However, these applications have largely remained proof-of-concepts. To our knowledge, no prior work has attempted to systematically deploy FL to construct a large-scale, contin-

uously evolving cognitive ecosystem for the complex, safety-critical domain of maritime navigation. Existing research has not addressed how to manage the significant statistical heterogeneity of data from vastly different maritime environments (e.g., Arctic vs. tropical, open ocean vs. cluttered ports), a challenge our framework directly confronts.

2.2. Transfer Learning and Foundation Models: Building upon Universal Priors

Training complex AI models from scratch is notoriously data-hungry and computationally expensive. Transfer Learning, particularly the pre-training and fine-tuning paradigm, has become a standard practice to mitigate this [11,12]. This approach has evolved into the modern concept of Foundation Models, such as BERT in natural language processing and Vision Transformers (ViT) in computer vision. These models are pre-trained on massive, diverse datasets, allowing them to learn rich, general-purpose representations and universal priors of their respective domains [13,14]. This pre-acquired “knowledge” provides a powerful starting point for a wide range of downstream tasks, dramatically reducing the data and time required for effective performance.

The concept is gaining traction in robotics, with a growing interest in developing “robotics foundation models” for general-purpose manipulation and navigation [15,16]. The idea of a “Sea-Sense” foundation model, pre-trained on global AIS trajectories, oceanographic data, and simulated scenarios, is a logical and powerful extension of this paradigm [17]. However, a critical open question remains: how can the universal knowledge of a foundation model be effectively and efficiently fused with the highly specific, private, and ever-changing experiences of individual, operational agents? A foundation model alone is static; it cannot learn from the ongoing, real-world experience of a deployed fleet. This gap highlights the need for a mechanism of continuous, collaborative learning. However, a foundation model alone is static; it cannot learn from the ongoing, real-world experience of a deployed fleet. This highlights the critical need for a mechanism of continuous, collaborative learning.

2.3. Meta-Learning: The “Last Mile” of Rapid, Personalized Adaptation

While foundation models provide a strong generic starting point, every physical agent, especially a USV, possesses unique characteristics—subtle differences in hydrodynamics, sensor calibration biases, or actuator responses. Fine-tuning for each individual vessel can still be a costly process. Meta-Learning, or “learning to learn,” directly addresses this challenge of rapid personalization [18]. Algorithms like Model-Agnostic Meta-Learning (MAML) and context-based approaches like RL^2 train a model on a distribution of diverse tasks, explicitly optimizing its ability to adapt to a new task with only a handful of examples (few-shot adaptation) [19,20].

In autonomous navigation, meta-learning has been successfully applied to help agents quickly adapt to new environmental parameters or dynamics [21], as in our precursor work. However, meta-learning is typically performed in a centralized manner on a curated set of training tasks. Its effectiveness is thus limited by the diversity of this pre-collected dataset. It does not, by itself, provide a mechanism for the model to benefit from the continuous stream of new experiences from a globally distributed fleet. It represents the “last mile” of adaptation, but requires a powerful, globally informed starting point to be truly effective. The synergy between FL and meta-learning, often termed Federated Meta-Learning (FML) [22], is an emerging research frontier, but its application to complex, embodied AI systems like USVs remains largely unexplored. However, meta-learning is typically performed in a centralized manner on a pre-collected dataset. Its effectiveness is thus limited by the diversity of this dataset and it does not, by itself, provide a mechanism for the model to benefit from the continuous stream of new experiences from a globally

distributed fleet. However, meta-learning is typically performed in a centralized manner on a pre-collected dataset. Its effectiveness is thus limited by the diversity of this dataset and it does not, by itself, provide a mechanism for the model to benefit from the continuous stream of new experiences from a globally distributed fleet.

2.4. Synthesis and Our Contribution: The Case for an Integrated FMTL Framework

In reviewing these three powerful learning paradigms, a clear picture emerges. Transfer Learning provides a universal foundation. Federated Learning offers a privacy-preserving pathway for collaborative, lifelong evolution. Meta-Learning enables rapid, low-cost personalization. Yet, they have largely been investigated in isolation. To date, no research has proposed a cohesive, hierarchical framework that systematically integrates all three to address the end-to-end lifecycle of an autonomous agent's intelligence—from its “birth” with foundational knowledge, through its “social learning” within a global community, to its “maturation” as a personalized, field-adapted expert [23–25].

This critical gap is precisely what our work aims to fill. The proposed Federated Meta-Transfer Learning (FMTL) framework is the first to architect a seamless synthesis of these paradigms. It leverages transfer learning for initialization, federated learning for collaborative and privacy-preserving knowledge sharing across fleets, and meta-learning for the final, rapid adaptation to individual agents. By doing so, our framework provides a holistic solution to the data silo dilemma and offers a concrete technical blueprint for realizing a truly scalable, ever-improving, and universally applicable maritime intelligence.

3. Materials and Methods

To address the challenge of balancing global optimality with local dynamic collision avoidance for Unmanned Surface Vehicles (USVs) in complex maritime environments, we propose a novel hierarchical decision-making framework called Theta-TPPO. This framework decouples the path planning task into two collaborative modules: a global planner responsible for generating smooth macro-level paths, and a local agent for real-time, rule-compliant dynamic decision-making. This section first presents the USV kinematic model and formal problem definition, then details the two core components of the Theta-TPPO framework: the Theta-based global path planner and the Transformer-enhanced PPO local decision agent.

To accurately simulate USV motion, we adopt a three degrees-of-freedom (3-DOF) model focusing on surge, sway, and yaw in the horizontal plane. The nomenclature and symbol definitions used in this study are summarized in Table 1.

Table 1. Nomenclature and Symbol Definitions.

Symbol	Definition	Dimension/Type
System Modeling		
$\eta = [x, y, \psi]^T$	USV position and heading in inertial frame	$\mathbb{R}^3$
$v = [u, v, r]^T$	USV linear and angular velocities in body frame	$\mathbb{R}^3$
$M, C(v), D(v)$	Inertia, Coriolis, and Damping matrices	3×3
$\tau_{control}, \tau_{env}$	Control forces and environmental disturbances	$\mathbb{R}^3$
Federated Learning		
ω_t	Global model weights at communication round t	$\mathbb{R}^d$
F_k, D_k	Participating Fleet k and its private dataset	-
Δ_k	Local model weight update (gradient) from Fleet k	$\mathbb{R}^d$
RL & Network		
s_t, a_t, r_t	State, Action, and Reward at timestep t	-
π_ϕ, V_ψ	Actor policy network and Critic value network	Neural Networks
$\vartheta = (v, \varepsilon)$	Dynamic Tactical Relational Graph	Graph Structure
h_{scene}	Latent scene embedding from Graph-Gated Transformer	$\mathbb{R}^{128}$

3.1. The Federated Meta-Transfer Learning (FMTL) Framework: An Overview

To realize our vision of a Global Maritime Cognitive Network, we introduce the Federated Meta-Transfer Learning (FMTL) framework, a hierarchical architecture designed to manage the entire lifecycle of an agent's cognitive development. This framework provides the high-level structure within which our specific navigation and decision-making modules operate. As illustrated in Figure 1, the FMTL framework consists of three distinct, sequential stages:

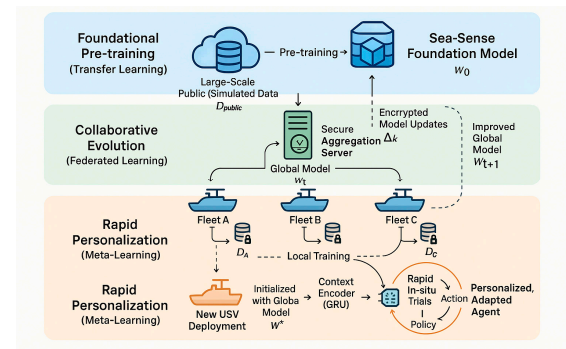


Figure 1. Federated Meta-Transfer Learning Framework.

3.1.1. Stage 1: Foundational Pre-Training via Transfer Learning

The starting point for all intelligence within our ecosystem is a universal “Sea-Sense” foundation model. This model, based on our Transformer-enhanced architecture (detailed in Section 3.3), is pre-trained on a massive-scale, generalized dataset comprising public maritime data (e.g., historical AIS tracks) and extensive, high-fidelity simulations. The objective of this stage is not to create a perfect navigator, but to instill the model with a robust set of universal priors, including the following:

Fundamental Physics: An implicit understanding of vessel kinematics and momentum.

General COLREG Compliance: A basic intuition for common encounter scenarios like head-on and crossing situations.

Scene Representation: The ability to encode complex, multi-object maritime scenes into a meaningful latent representation.

This pre-training process leverages the power of transfer learning to provide a powerful, knowledge-rich initialization for all subsequent learning, significantly accelerating convergence and improving generalization.

3.1.2. Stage 2: Collaborative Evolution via Federated Learning

This stage addresses the core challenge of privacy-preserving, cross-fleet knowledge sharing. The pre-trained foundation model is distributed to a set of N participating, non-trusting fleets, $F = F_1, F_2, \dots, F_N$. The training proceeds in communication rounds, $t = 1, 2, \dots$. In each round:

Distribution: A central aggregation server distributes the current global model weights, w_t , to all participating fleets.

Local Training: Each fleet F_k trains the model on its own private, local dataset D_k for a number of epochs, resulting in a set of locally updated model weights, w_{t+1}^k . This local training utilizes the T-PPO algorithm (detailed in Section 3.4).

Secure Aggregation: Each fleet computes its local weight update, $\Delta_k = w_{t+1}^k - w_t$. These updates, rather than the raw data, are then encrypted and sent to the central server.

The server performs a secure aggregation, typically a weighted average, to compute the global model update for the next round:

$$w_{t+1} = w_t + \sum_{k=1}^N (n_k/n) * \Delta_k \quad (1)$$

where n_k is the size of the local dataset D_K and n is the total size of all data.

This federated learning cycle allows the global model to continuously learn from the diverse, real-world experiences of all participating fleets, creating a shared intelligence far more powerful than any single fleet could develop in isolation. Beyond this collective capability, the design is specifically engineered to address three fundamental constraints of the maritime domain:

Communication Efficiency: Transmitting raw sensor data via maritime satellite (VSAT) is often cost-prohibitive. By exchanging only model parameters (~150 MB) rather than terabytes of raw telemetry, we reduce bandwidth usage by orders of magnitude, ensuring viability over low-bandwidth links.

Handling Heterogeneity: Participating fleets operate in diverse environments (Non-IID data), such as open oceans vs. congested ports. The weighted aggregation mechanism (Equation (1)) ensures that the global model effectively balances these disparate domains without being biased towards any single fleet's distribution.

Security and Privacy Assumptions: To enable collaboration between potentially competing commercial entities, the framework operates on a strict "data sovereignty" model. We assume that raw operational logs remain exclusively on-premise, with all inter-fleet model transmissions secured via TLS encryption to prevent interception.

Given the harsh nature of maritime communications and the potential for adversarial interference, our FL protocol incorporates specific mechanisms to ensure system resilience:

1. Resilience to Node Failures (Stragglers):

Maritime fleets frequently experience intermittent connectivity (e.g., satellite blockage). To prevent the entire training process from stalling due to a single unresponsive vessel ("straggler"), we implement a Synchronous Aggregation with Timeout protocol. The server waits for a specified time window τ for updates. If a fleet F_k fails to upload its update Δ_k within this window, it is excluded from the current round's aggregation (Equation (1)). The global model updates based on the subset of surviving nodes. The "dropped" fleet retains its local state and attempts to re-sync with the updated global model in the subsequent round, ensuring that temporary connection losses do not halt the collective evolution.

2. Defense Against Malicious Participants:

To mitigate the risk of "Model Poisoning" attacks—where a compromised fleet uploads corrupted gradients to degrade the global model—we employ two defensive strategies:

Update Norm Clipping: Before aggregation, the server checks the L2-norm of each incoming update $\|\Delta_k\|_2$. Updates exceeding a dynamic threshold γ are scaled down: $\Delta_k \leftarrow \Delta_k / \max(1, \|\Delta_k\|_2 / \gamma)$. This limits the impact any single malicious actor can have on the global weight manifold.

Anomaly Detection (Robust Aggregation): While Equation (1) uses a weighted average, our framework supports Trimmed Mean aggregation in high-risk scenarios. By discarding the top and bottom $b\%$ of values for each parameter dimension before averaging, the system effectively filters out statistical outliers caused by Byzantine faults or intentional attacks, provided that the majority of fleets remain honest.

3.1.3. Stage 3: Rapid Personalization via Federated Meta-Learning

The final stage enables the rapid and efficient deployment of a new vessel. A new agent is initialized with the latest, most powerful global model w from the federated network. This model serves as an exceptional knowledge-rich starting point for a meta-learning process.

We adopt a context-based meta-learning approach (as detailed in Section 3.5), where the powerful federated model w provides the initial policy parameters. The agent then performs a small number of K “adaptation trials” (e.g., $K = 5$ to 10) in its specific deployment environment. During these trials, the agent’s recurrent context encoder (GRU) rapidly infers the unique latent characteristics of its new embodiment (e.g., a more sluggish vessel) and environment (e.g., strong currents). This allows the policy to be modulated in-context without requiring substantial weight updates, achieving near-optimal performance with minimal interaction. This final step ensures that the globally informed intelligence is perfectly tailored to the local, individual reality of each vessel. The technical details of our context-based meta-learning approach are presented in Section 3.5.

3.2. System Modeling and Problem Formulation

3.2.1. Kinematic Equations

We adopt the standard 3-Degrees-of-Freedom (3-DOF) maneuvering model described by Fossen [26], assuming that the USV is a rigid body operating on the sea surface where heave, roll, and pitch motions are negligible. The USV state in the North-East-Down (NED) frame is defined by the pose vector $\mathbf{v} = [u, v, r]^T$ and the velocity vector in the body-fixed frame $\mathbf{v} = [u, v, r]^T$. The kinematics are governed by the following:

$$\dot{\eta} = R(\psi)\mathbf{v} \quad (2)$$

where $R(\psi)$ is the rotation matrix:

$$R(\psi) = [\cos(\psi), -\sin(\psi), 0; \sin(\psi), \cos(\psi), 0; 0, 0, 1] \quad (3)$$

3.2.2. Hydrodynamic and Stochastic Modeling

To enhance realism, we incorporate simplified hydrodynamic damping and sensor noise. The generalized force is calculated as follows:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{control} - D\mathbf{v} \quad (4)$$

Specifically, $D = \text{diag}(Xu, Yv, Nr)$ represents the hydrodynamic linear damping coefficients. The system dynamics also implicitly account for the inertia matrix $M = \text{diag}(m - u, m - Yv, Iz - Nr)$, which incorporates both the rigid-body mass and the hydrodynamic added mass terms [26], ensuring the fidelity of the simulation.

where D is the linear damping matrix. Furthermore, the discrete-time motion update includes stochastic noise terms to simulate sensor uncertainty:

$$\begin{cases} x_{t+1} = x_t + (u_t \cos(\Delta t + \varepsilon_x)) \\ y_{t+1} = y_t + (u_t \sin(\Delta t + \varepsilon_y)) \\ \psi_{t+1} = \psi_t + r_t \Delta t + \varepsilon_\psi \end{cases} \quad (5)$$

where $\varepsilon_x, \varepsilon_y \sim \mathcal{N}(0, \sigma_p^2)$ and $\varepsilon_\psi \sim \mathcal{N}(0, \sigma_h^2)$. All key parameters for the simulation are summarized in Table 2.

Table 2. USV Simulation Parameters.

Parameter	Symbol	Value	Description
Surge Damping	X_u	100 N·s/m	Linear damping coefficient
Sway Damping	Y_v	200 N·s/m	Linear damping coefficient
Yaw Damping	N_r	50 N·m·s/rad	Rotational damping coefficient
Position Noise Std. Dev.	σ_p	0.1 m	Simulates GPS error
Heading Noise Std. Dev.	σ_h	0.5°	Simulates compass error
Simulation Timestep	Δt	0.1 s	Integration interval

Within our hierarchical framework, given a map containing static obstacles, a start point S , and a goal point G , the global planner first generates a sequence of waypoints $W = wp_1, wp_2, \dots, wp_M$. Subsequently, the local decision agent decomposes this into sub-tasks, sequentially treating each waypoint wp_1 as a temporary target while avoiding dynamic obstacles and complying with COLREGs in real time.

3.3. Global Path Planner: Theta Algorithm

To generate efficient and smooth global paths, we employ the Theta algorithm [27], a significant improvement over traditional A [28]. While standard A searches on grid maps producing paths constrained to grid directions (resulting in unnecessary turns and suboptimal Euclidean distances), Theta overcomes this limitation through a line-of-sight checking mechanism.

During Theta's search process, when expanding from node p to neighbor node s , the algorithm applies the following update rules:

- Standard path cost calculation: $c(s) = g(p) + \text{cost}(p, s)$.
- Line-of-sight check: Verify if obstacles exist between s and parent(p).
- Path update:
 - (a) If line-of-sight exists between s and parent(p), calculate shorter path cost: $c'(s) = g(\text{parent}(p)) + \text{cost}(\text{parent}(p), s)$.
 - (b) If $c'(s) < c(s)$, set parent of s directly to parent(p) and update $g(s) = c'(s)$.
 - (c) Otherwise, maintain parent of s as p .

This “any-angle” path straightening enables Theta to generate paths unconstrained by grid topology, producing trajectories that are 5–10% shorter than standard A while maintaining comparable computational complexity. The algorithm outputs sparse but critical waypoints W that serve as macro-level guidance for the local agent.

3.4. Local Decision Agent: Cognitive Navigation via Graph-Gated Transformer PPO (GGT-TPPO)

While the FMTL framework provides the macro-level architecture for collaborative learning, the micro-level intelligence of each agent is driven by a sophisticated local decision agent. This agent is responsible for real-time, rule-compliant dynamic collision avoidance while tracking global waypoints. We move beyond standard “flat” perception models by introducing the Graph-Gated Transformer-enhanced Proximal Policy Optimization (GGT-TPPO) agent. The detailed architecture of the proposed GGT-TPPO local decision agent is illustrated in Figure 2. This architecture's core innovation is its ability to perform structured relational reasoning, transforming raw perceptual data into a deep, causal understanding of the local maritime scene. While the concept of graph-based reasoning is established in other domains, our core contribution in the design of the GGT-TPPO agent is twofold and specifically tailored to the maritime context. First, we propose a domain-specific method for constructing the Dynamic Tactical Relational Graph (TRG), where edges are not implicitly learned but are dynamically generated using heuristic rules derived directly from maritime regulations (COLREGs) and kinematic principles (i.e., TCPA/DCPA thresholds). This injects critical expert knowledge into the model, forcing it to reason over a semantically

rich and causally relevant structure. Second, to our knowledge, this work represents the first integration of a Graph-Gated Transformer (GGT) mechanism with a Proximal Policy Optimization (PPO) agent to solve the complex task of multi-vessel, COLREGs-compliant dynamic collision avoidance for USVs. This fusion of macro-level federated learning with a micro-level structured reasoning agent is the cornerstone of our framework's success, moving beyond "flat" perception models to enable a deeper, more interpretable, and robust decision-making process.

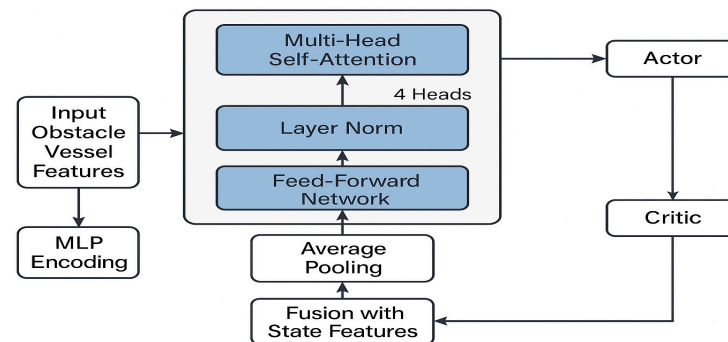


Figure 2. Transformer Encoder Architecture.

The GGT-TPPO agent models this task as a Markov Decision Process (MDP), but with a crucial enhancement: before making a decision, it first constructs a dynamic relational graph to represent the underlying tactical structure of the environment.

3.4.1. Dynamic Tactical Relational Graph (TRG) Construction

Instead of treating all perceived objects as an unordered set, our agent first organizes them into a Dynamic Tactical Relational Graph (TRG), $G_t = (V, E_t)$. This graph serves as a powerful, explicit prior that guides the model's reasoning process.

Nodes (V): The nodes of the graph are the feature vectors of all entities perceived by the ego-USV at timestep t , including itself, other vessels, and static obstacles. Each node v_i represents an object's state (relative position, velocity, etc.).

Edges (E_t): The edges are the core of our reasoning structure. They are not learned implicitly but are dynamically generated at each timestep based on a set of domain-specific heuristics derived from maritime regulations (COLREGs) and physical principles. An edge (i, j) is created if it meets one of the following criteria:

Collision-Risk Edge: An edge is formed if the calculated Time to Closest Point of Approach (TCPA) and Distance at Closest Point of Approach (DCPA) between entities i and j cross a critical safety threshold. This directly encodes imminent collision threats into the graph structure.

Situational-Awareness Edge: Edges are created based on the relative bearing between the ego-vessel and other vessels, explicitly representing key COLREGs encounter types such as Head-on, Crossing, and Overtaking. Each edge can be typed to reflect the specific nature of the encounter.

Proximity Edge: A general-purpose edge is formed if the Euclidean distance between i and j is within a defined perception radius, capturing general spatial relationships.

This process transforms a flat, unstructured scene into a sparse, semantically rich graph that highlights the most critical relationships for safe and efficient navigation.

3.4.2. Graph-Gated Transformer (GGT) for Structured Reasoning

The TRG is then used to structure the reasoning process of a Transformer encoder. This is achieved through a Graph-Gated Attention mechanism. Unlike a standard Transformer

that performs all-to-all attention, the GGT constrains the flow of information to only traverse the edges of the TRG.

Given the sequence of node features X , the standard self-attention score between nodes i and j is modulated by an attention bias matrix M_{graph} derived directly from the TRG's adjacency matrix A_t :

$$\text{Attention}(Q, K, V) = \text{softmax}((Q * K^T \sqrt{D_k}) + M_{\text{graph}}) * V \quad (6)$$

where $(M_{\text{graph}})_{ij} = 0$ if an edge exists between i and j in A_t , and $(M_{\text{graph}})_{ij} = -\infty$ otherwise.

This hard-gating mechanism has a profound effect: it forces the agent to focus its cognitive resources exclusively on the relationships deemed critical by the TRG. It learns not just to see other vessels, but to reason about its specific relationship to them (e.g., “This is a vessel I am on a collision course with,” or “This is a vessel in a crossing situation from my starboard side”). The output of the multi-layer GGT is a context-aware and, more importantly, relationally grounded embedding of the entire maritime scene, h_{scene} .

3.4.3. Decision-Making and MDP Formulation

The final scene embedding h_{scene} is concatenated with the agent's own state and task information (e.g., distance to next waypoint) to form the complete state representation for the PPO agent. The rest of the MDP formulation follows the standard procedure:

State Space (S): The state s_t includes the GGT's scene embedding h_{scene} , the ego-USV's state (S_{ego}), and the task state (S_{task}).

Action Space (A): A continuous 2D action $a_t = [\delta, \tau]^T$ representing rudder angle and thrust commands.

Reward Function: The multi-objective reward function guides the agent towards safe, efficient, and COLREGs-compliant behavior.

By integrating the GGT, our T-PPO agent learns a far more robust and interpretable policy. Its superior performance, especially its rapid adaptation in new environments during the FMTL Stage 3, stems from its ability to quickly infer the underlying causal and relational structure of a novel scene, rather than merely pattern-matching on superficial features. This fusion of macro-level federated learning and micro-level structured reasoning is the cornerstone of our framework's success. The reward function components and their corresponding parameters are summarized in Table 3.

Table 3. Reward Function Parameters.

Component	Weight/Constant	Value	Purpose
Waypoint Reward	w_1	0.8	Encourage goal-directed navigation
Safety Reward	w_2	2.0	Prioritize collision avoidance
COLREGs Reward	w_3	2.5	Maximize rule compliance
Smoothness Penalty	w_4	0.1	Encourage energy-efficient control
Progress Constant	C_1	1.0	Scale waypoint progress reward
Control Constant	C_2	0.01	Scale control penalty
Collision Penalty	-	-1000	Terminate episode on collision

The COLREGs-compliant reward mechanism for different encounter scenarios is illustrated in Figure 3.

3.4.4. T-PPO Network Architecture

To process complex environmental states containing multiple dynamic vessels, we design a Transformer-based PPO network architecture, as shown in Figure 4.

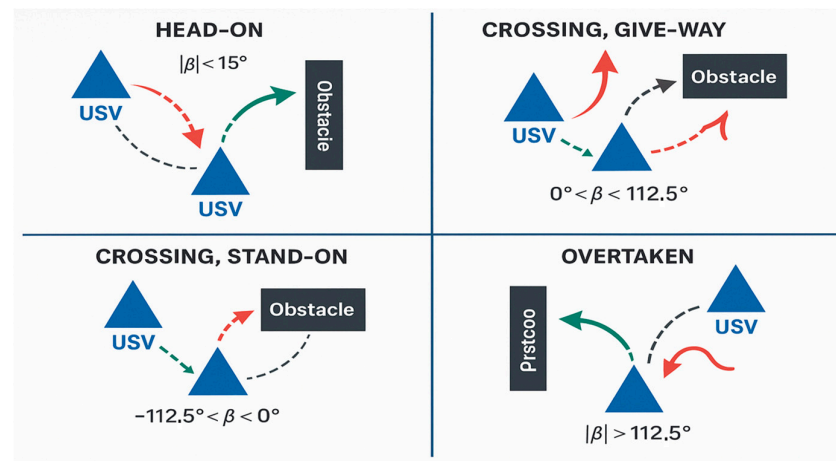


Figure 3. COLREGs-Compliant Reward Mechanism.

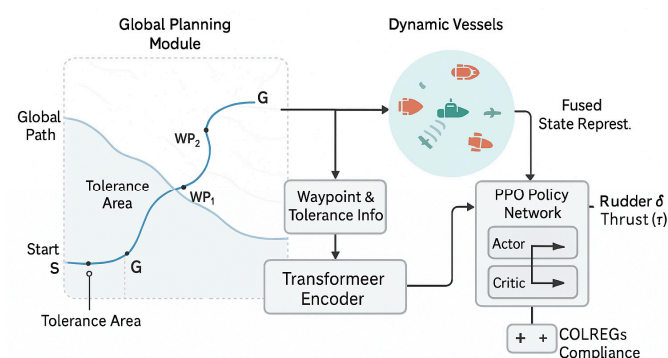


Figure 4. Hierarchical Interaction Flow of the Theta-TPPO Framework.

Input Encoding:

- Own-ship and task states are concatenated and encoded through MLP_{state} (2 layers, 128 neurons each) to produce state feature $z_{state} \in R^{128}$.
- Each obstacle feature vector o_k is mapped to high-dimensional space through MLP_{obs} (2 layers, $64 \rightarrow 128$ neurons) to produce obstacle embeddings $e_1, \dots, e_k$, where $e_k \in R^{128}$.
- Transformer Encoder: The obstacle embeddings are processed by a compact Transformer encoder configuration:
 - 2 Transformer layers with 4 attention heads per layer.
 - Model dimension $d_{model} = 128$, Feed-forward dimension $d_{ff} = 512$.
 - Layer normalization and residual connections.
 - No positional encoding (as vessel set is unordered).

A compact configuration of 2 layers and 4 attention heads was found to be sufficient for capturing the necessary vessel interactions while maintaining computational efficiency for real-time inference. The self-attention mechanism is formulated as follows:

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k})V \quad (7)$$

where queries Q , keys K , and values V are linear projections of the input embeddings, and $d_k = 128$. Multi-head attention enhances representation capacity:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \quad (8)$$

where $h = 4$ heads compute attention with separate learned projections.

Information Aggregation and Decision:

- Transformer outputs are aggregated via average pooling to form environment context vector $c_{env} \in R^{128}$.
- State feature Z_{state} and environment context c_{env} are concatenated to form final fused representation $z_{fused} \in R^{128}$.
- Both Actor and Critic networks consist of 3 fully connected layers (256→output) with ReLU activation.
- Actor outputs Gaussian distribution parameters for continuous action sampling.
- Critic outputs scalar state value estimate.

The overall enhanced hybrid framework architecture, integrating global planning, local decision-making, and COLREGs compliance, is illustrated in Figure 5.

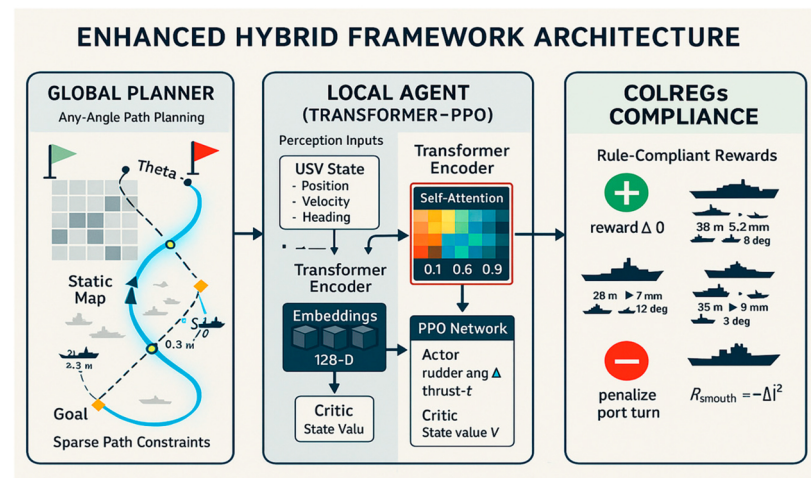


Figure 5. Enhanced Hybrid Framework Architecture.

This architecture enables the agent to dynamically weight the importance of different obstacle vessels, a critical capability that traditional MLP-based approaches lack. The network structure of the proposed T-PPO agent is illustrated in Figure 6. The mathematical foundation ensures that the model can learn complex, non-linear interaction patterns while maintaining computational efficiency suitable for real-time deployment.

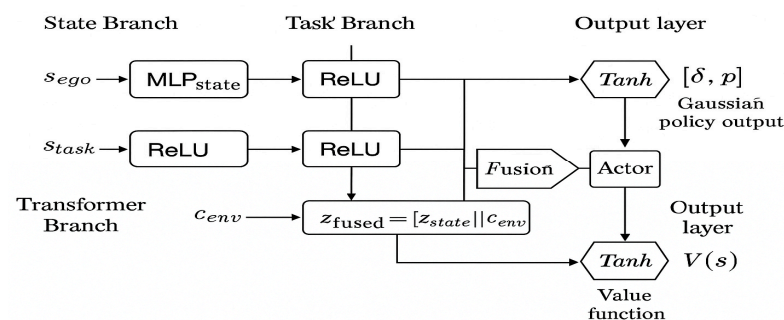


Figure 6. T-PPO Network Structure.

3.5. Meta-Reinforcement Learning for Fast Adaptation

Nomenclature: Throughout this section, we refer to the full meta-learning framework as Meta-Theta-TPPO and to the underlying hybrid decision architecture as Theta-TPPO. The integrated meta-learning and safety-shield architecture of the proposed Meta-Theta-TPPO framework is illustrated in Figure 7. The local policy used within the hybrid architecture is the Transformer-enhanced PPO agent (T-PPO) described in Section 3.3.

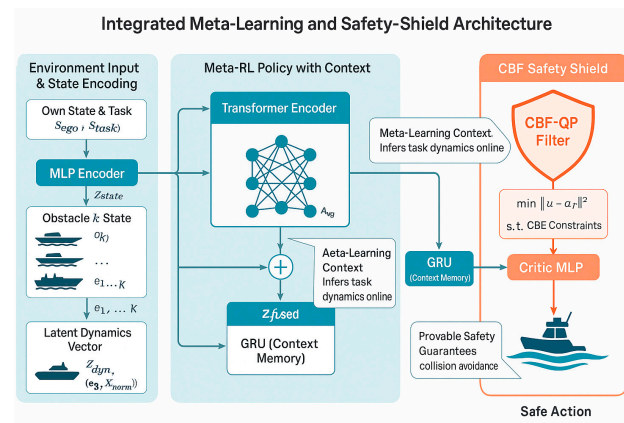


Figure 7. Integrated Meta-Learning and Safety Shield Architecture.

3.5.1. Problem Formulation

In our framework, a task $T_i \sim p(T)$ denotes a concrete instantiation of the maritime environment, characterized by (i) ego-vessel dynamics, (ii) heterogeneous multi-vessel behavior patterns, and (iii) environmental conditions (currents, sea state). The policy π_θ must quickly adapt its parameters to θ using a small adaptation dataset D_i^{adapt} (e.g., one episode or a few trials) from T_i .

We seek initial parameters θ^* that maximize post-adaptation performance:

$$\theta^* = \arg \max_{\theta} E_{T_i \sim p(T)} [J_{T_i}((D_i^{adapt}, \theta))] \quad (9)$$

where U is the adaptation operator, and $J_{T_i}(\cdot)$ is the expected return on task T_i after adaptation.

Parameter conventions. We use θ for meta-initialization parameters (outer-loop) and ϕ for the working policy parameters used in deployment/training code. In practice, $\phi \leftarrow \theta$ at the start of each task and is adapted by U .

3.5.2. Task Distribution Design (Enhanced)

To induce robust few-shot adaptation, we sample tasks by varying the following:

- **Ego-Vessel Dynamics.** Hydrodynamic coefficients are drawn from predefined ranges spanning agile to sluggish vessels. These values are encoded and passed to the policy as the latent dynamics vector Z_{dyn} creating an explicit physical context for adaptation.
- **Heterogeneous Traffic Behaviors.** Other vessels are instantiated from behavior archetypes (e.g., compliant, aggressive, erratic), altering encounter patterns and COLREGs contexts across episodes.
- **Environmental Conditions.** Currents/sea state magnitudes and directions are randomized to produce distinct disturbance profiles.
- **Link to architecture.** The Z_{dyn} vector is concatenated with perception features (cf. Equation (6)) so that meta-learning must learn how to use explicit dynamics information for rapid in-context adjustment.

3.5.3. Context-Based Meta-RL (RL²-Style) with GRU Integration

We adopt a context-based meta-RL approach (RL²-style). Concretely, we augment Theta-TPPO with a GRU placed after the Transformer encoder and feature-fusion stage (producing Z_{fused}) and before the Actor/Critic MLP heads. At time t , the network input is $z_{fused,t}, z_{dyn}, a_{t-1}, r_{t-1}$ and the GRU hidden state h_t accumulates recent interaction context.

The policy/value functions are then conditioned on both current perceptual features and the inferred context:

$$a_t \sim \pi_\phi(z_{fused,t}, z_{dyn}, h_t) \quad (10)$$

$$V_\psi = V_\psi(z_{fused,t}, z_{dyn}, h_t) \quad (11)$$

This design enables the agent to implicitly infer latent task parameters (e.g., responsiveness of our vessel; aggressiveness of traffic) within a few trials and adjust its behavior accordingly.

3.6. Provable Safety via Control Barrier Function (CBF) Shield

To provide formal safety guarantees during adaptation and deployment, every policy action is passed through a CBF-based safety filter that minimally modifies the action only when necessary.

3.6.1. Modeling for Safety Design

For safety analysis, we abstract the USV kinematics/dynamics into a control-affine continuous-time model:

$$\dot{x} = f(x) + g(x)u \quad (12)$$

where $\dot{x}$ includes position and heading and u is the control input. This model approximates the more detailed stochastic simulation used for training and evaluation (Section 3.1), enabling efficient, real-time constraint computation while preserving essential vessel behavior.

3.6.2. Safe Set and CBF Constraint

For each obstacle j , define the safety function $h_j(x)$ (e.g., squared inter-vessel distance). The safe set is

$$C = \{x \mid h_j(x) \geq D_{safe}^2, \forall j\}. \quad (13)$$

The standard CBF condition that renders C forward invariant is as follows:

$$L_f B_j(x) + L_g B_j(x)u + \alpha(B_j(x)) \geq 0, \forall j \quad (14)$$

with L_f, L_g denoting Lie derivatives and $\alpha(\cdot)$ a class-K function (e.g., $\alpha(B) = \gamma$).

3.6.3. Quadratic-Program (QP) Safety Filter and Integration

Given the T-PPO action proposal u_{rl} , the safety shield solves in real time:

$$\begin{cases} u^* = \arg \min_u \|u - u_{rl}\|_2^2 \\ \text{s.t. } L_f B_j(x) + L_g B_j(x)u + \alpha(B_j(x)) \geq 0, \forall j, \\ u \in U \text{ (actuator bounds)} \end{cases} \quad (15)$$

Remark on discrete updates. Although control is issued in discrete timesteps Δt , trigonometric/kinematic updates place Δt outside of sines/cosines (e.g., $u_t \cos \psi_t \cdot \Delta t$), while safety is computed from the continuous-affine abstraction for tractability.

3.7. Integrated Framework and Implementation Details

This section consolidates practical details that tie global planning, local decision-making, meta-adaptation, and safety into a single integrated online framework.

- **Global–Local Interface.** Theta provides sparse, kinematically favorable waypoints W . The local T-PPO agent tracks the current waypoint while performing dynamic, COLREGs-compliant avoidance.

- **Action Safety.** Each policy output is filtered through the CBF-QP to guarantee collision avoidance with minimal intervention.
- **Real-Time Optimizations.** We employ waypoint pre-culling, Transformer encoding caches for static obstacles, and an asynchronous perception–decision–execution pipeline to achieve sub-10 ms decision cycles (details preserved in Section 4).
- **Failure Recovery.** A disaster recovery routine monitors a composite risk metric (DCPA/TCPA with temporal decay) and can trigger emergency stop/maneuvers and mission abort logic when necessary (see Algorithm 1 notes).

Algorithm 1. Federated Learning Loop for the Global Maritime Cognitive Network

Goal: Learn a powerful global model w by leveraging private data from N fleets.

1: Initialize: Server initializes global model with foundation model weights w_0 .

2: for each communication round $t = 1, 2, \dots, T$ do

3: Distribution: Server sends current global model W_t to a set of selected fleets S_t

4: for each fleet k in S_t in parallel do

5: // Local Fleet Training

6: $w_{t+1}^k \leftarrow \text{LocalFleetUpdate}(k, w_t)$ // This step calls Algorithm 2

7: end for

8: Aggregation: Server collects all local updates w_{t+1}^k from the fleets.

9: $w_{t+1} \leftarrow$ Securely aggregate the updates (e.g., using Federated Averaging).

10: end for

11: return final global model w_T .

Function $\text{LocalFleetUpdate}(k, w_t)$:

1: Client k sets its local model weights to w_t .

2: Train the local model on its private dataset D_k for E local epochs using the T-PPO Agent Training (Algorithm 2).

3: return the updated local model weights w_{t+1}^k .

Algorithm 2. Theta-TPPO Hybrid Decision Framework (Deployment, with Safety Shield)

Input: Static map M_{static} , start S , goal G ; pre-trained T-PPO policy π_ϕ .

Output: USV trajectory.

$W \leftarrow \text{Theta}(M_{static}, S, G)$

Set $i \leftarrow 0$, $current_{target} \leftarrow W[i]$, $emergency_{count} \leftarrow 0$

while goal not reached do

 |— Perceive own-ship/task/obstacles; build s_t .

 |— Compute collision risk $R(s_t)$ (DCPA/TCPA with decay).

 |— if $R(s_t) > \text{CRITICAL}$ then

 |— EmergencyStop(); $emergency_{count} += 1$.

 |— if $emergency_{count} > \text{MAX ATTEMPTS}$ then

 |— return MISSION ABORTED.

 |— $a_t \leftarrow \text{Emergency Maneuver}(s_t)$.

 |— else

 |— $a_t \leftarrow \pi_\phi(s_t)$ // policy proposal

 |— $a_t \leftarrow \text{CBF}_{Shield}(s_t, a_t)$ // Safety filter (this work)

 |

 |— Execute a_t ; update state; advance waypoint if within threshold.

 |— Timeout guard; possibly return MISSION TIMEOUT.

end while

return MISSION SUCCESS.

4. Training Methodology and Algorithms

This chapter consolidates all the pseudocode. Algorithm 1 presents the Theta-TPPO deployment loop with the CBF safety line explicitly integrated. Algorithm 2 details T-

PPO training, and Algorithm 3 provides the Meta-Training Loop for Meta-Theta-TPPO (outer-loop over tasks with context-based inner adaptation).

Algorithm 3. T-PPO Agent Training (Optimization-Aware)

```

Initialize: Actor  $\pi_\phi$ , Critic  $V_\psi$  (Transformer encoder + GRU context as in Section
3.4.3), PPO hyperparameters  $\{\epsilon, \gamma, \lambda\}$ , replay buffer  $D$ , static cache  $C_{static}$ .
Pre-compute and cache static obstacle embeddings into  $C_{static}$ .
for episode = 1 ...  $N_{episodes}$  do
    |—— Reset environment; clear dynamic cache.
    |—— for  $t = 0 \dots T_{max} - 1$  do
        |—— Build encoded state using cached static context + current dynamic en-
codings.
        |    |—— Sample  $a_t \sim \pi_\phi(\cdot | s_t)$ ; step env; observe  $r_t, s_{t+1}$ .
        |    |—— If emergency triggered, apply additional penalty to  $r_t$ .
        |    |—— Store  $(s_t, a_t, r_t, s_{t+1})$  in  $D$ .
        |
        |    if update interval reached then
            |—— Compute GAE advantages  $\hat{A}$  and returns  $R$ .
            |—— for epochs = 1 ...  $N_{epochs}$  do
                |—— Maximize clipped PPO objective for  $\pi_\phi$ .
                |—— Minimize value loss for  $V_\psi$ .
            |—— Clear  $D$ ; update caches if needed.
        |
        |——  $s_t \leftarrow s_{t+1}$ .
    end for
end for

```

4.1. The FMTL End-to-End Training and Deployment Pipeline

The training and deployment of our framework follow the three-stage process outlined in our Federated Meta-Transfer Learning (FMTL) architecture. This section consolidates the pseudocode and methodology for this entire pipeline, illustrating how a universally capable and locally adapted agent is forged.

4.1.1. Stage 1: Foundational Pre-Training (Transfer Learning)

The initial step is to build the “Sea-Sense” foundation model. This is a one-time, offline process performed before the federated learning commences.

Objective: To learn a general-purpose representation of maritime scenes and basic navigation priors.

Dataset: A large-scale, diverse dataset D_{public} is used, comprising either publicly available maritime data (e.g., historical AIS tracks fused with environmental data) or, as in our experiments, data generated from a wide distribution of high-fidelity simulations covering countless environmental and traffic scenarios.

Methodology: A standard supervised learning or reinforcement learning process is applied. The T-PPO agent (as detailed in Algorithm 2) is trained on D_{public} until its performance on a held-out validation set converges. The resulting model weights, w_0 , serve as the initial global model for the federated network.

4.1.2. Stage 2: Collaborative Training (Federated Learning)

This is the core iterative process for evolving the global model. We present the overarching federated learning loop in Algorithm 4.

Algorithm 4. Meta-Training Loop for Meta-Theta-TPPO (RL²-Style)

Goal: Learn initialization θ such that π_ϕ adapts rapidly on new tasks using recurrent context.

Initialize meta-parameters θ for Actor-Critic (including Transformer + GRU).

repeat (outer loop over meta-iterations)

- | — Sample a batch of tasks $\{T_i\} \sim p(T)$.
- | — for each T_i in batch do
 - | | — Set $\phi \leftarrow \theta$ (task start); reset GRU state h_0 .
 - | | — Collect adaptation trajectories D_i^{adapt} on T_i using recurrent policy
 - | | | with inputs $(z_{fused}, z_{dyn}, a_{t-1}, r_{t-1})$.
 - | | — Inner update (context-based): update hidden state across steps;
 - | | | optionally perform a small gradient step on ϕ using D_i^{adapt} (if hybrid RL² + gradient).
 - | | — Evaluate post-adaptation return J_{T_i} on held-out rollouts.
- | — Aggregate task-level policy/value losses (post-adaptation) across T_i .
- | — Meta-update $\theta \leftarrow \theta - \beta \nabla_{\theta} \Sigma_i L_{PPO}^i$ using PPO-style gradients
 - | | computed after adaptation behavior, keeping the recurrent context flow intact.

until convergence

return θ .

As shown in Algorithm 1, the federated process orchestrates the local training conducted by each fleet. The core of this local training is the T-PPO agent training procedure, which remains identical to what is described in your original Algorithm 3. This modularity is a key strength of our framework.

4.1.3. Stage 3: Deployment and Personalization (Federated Meta-Learning)

Once a robust global model w has been trained through the federated process, it can be deployed to any new vessel for rapid personalization. This stage leverages the meta-learning capabilities of the agent architecture. The methodology follows your original Algorithm 4 (Meta-Training Loop), with one critical difference: the initialization step.

Key Modification to Algorithm 3: Instead of initializing the meta-parameters θ from scratch (or randomly), they are initialized with the weights of the powerful, pre-trained global federated model w .

Initialize meta-parameters θ for Actor-Critic $\leftarrow w$

This modification transforms the process from standard meta-learning into Federated Meta-Learning. The agent is no longer “learning to learn” from a limited set of pre-defined tasks, but is “learning to rapidly personalize” from a starting point that already encapsulates the collective wisdom of an entire global network. This drastically accelerates the adaptation process described in Algorithm 4 and leads to a significantly higher final performance ceiling.

Notes: Risk R uses an exponentially decayed DCPA/TCPA fusion; emergency logic resets after safe operation; actuator bounds are enforced within the QP in line 11.

Notes: The training pipeline mirrors deployment optimizations (cache, windowed waypoints) to prevent train–test drift. Emergency trigger penalties encourage proactive avoidance that reduces reliance on recovery logic.

Notes: The GRU realizes in-context adaptation; optional small gradient steps within task episodes can be included (hybrid scheme) but are not required for RL²-style adaptation.

4.2. Real-Time Optimization Strategies (Kept and Clarified)

We retain the three-stage strategy:

1. **Waypoint pre-culling:** Remove waypoints that are too far or occluded to reduce processing overhead.
2. **Transformer encoding cache:** Pre-compute and cache embeddings for static obstacles, with selective refresh only for slow-moving entities.
3. **Asynchronous perception–decision–execution pipeline:** Decouple perception, decision-making, and execution to maximize throughput.

These optimizations reduce decision latency from ~12.8 ms to ~4.5 ms per cycle while preserving accuracy.

4.3. Implementation Complexity Analysis (Clarified)

Per-step complexity is dominated by attention and MLP passes. Caching reduces average-case cost by limiting re-computation to changed entities, with memory overhead in the order of the following:

$$O(N_{static} \cdot d + K \cdot d \cdot \tau) \quad (16)$$

where:

- N_{static} is the number of static obstacles.
- d is the embedding dimension.
- K is the number of cached dynamic entities.
- τ is the temporal window size.

Typical memory overhead is approximately ~50 MB for standard maritime scenarios.

5. Experiments and Results

To rigorously evaluate our Federated Meta-Transfer Learning (FMTL) framework, we designed a comprehensive suite of experiments. Our primary goal is twofold: first, to demonstrate that the federated paradigm effectively overcomes the “data silo” problem, creating a synergistic intelligence superior to any isolated agent; and second, to prove that our complete FMTL pipeline delivers unprecedented efficiency and performance during the deployment of new, unseen vessels.

5.1. Experimental Setup: Simulating a Federated World

We constructed a simulated ecosystem to mirror the core challenges of our vision: data fragmentation and operational heterogeneity. This ecosystem consists of three distinct, non-communicating fleets, each possessing its own private data distribution.

Fleet A (Open Ocean Specialists): Trained exclusively in scenarios with sparse, high-speed traffic, focusing on long-range COLREG encounters. Its private dataset is denoted as D_A .

Fleet B (Cluttered Port Navigators): Trained entirely within congested harbor environments characterized by numerous static obstacles, tight channels, and unpredictable vessel movements. Its dataset is D_B .

Fleet C (Adversarial Environment Survivors): Trained in scenarios with a high density of non-compliant and aggressive vessels, specializing in worst-case, safety-critical maneuvers. Its dataset is D_C .

“Sea-Sense” Foundation Dataset: Procedural Generation Protocol

To ensure the “Sea-Sense” foundation model acquires a universal prior, we developed a procedural scenario generation engine that creates the 50,000 unique scenarios

used in Stage 1. Unlike simple randomization, our generator uses a Constraint-Satisfying Configuration approach to enforce specific topological and tactical complexities.

The generation process is governed by the parameters detailed in Table 4 and follows the logic presented in Algorithm 5. The scenarios are categorized into three difficulty tiers to ensure curriculum learning:

Table 4. Procedural Generation Parameters for “Sea-Sense” Dataset.

Parameter Category	Parameter Name	Range/Value	Distribution
Environment	Map Dimensions	1000 m × 1000 m	Fixed
	Static Obstacles (Count)	0~30	Uniform Int
	Obstacle Size (Radius)	10 m~50 m	Gaussian ($\mu = 20, \sigma = 10$)
Ego Vessel	Current Velocity	0~1.5 m/s	Rayleigh
	Start Position	Edge of Map (Rand)	Uniform
	Goal Position	Opposite Edge (>800 m)	Constraint-based
Dynamic Traffic	Vessel Count	2~25	Poisson ($\lambda = 12$)
	Vessel Speed	5~20 knots	Uniform
	Interaction Type	Head-on, Crossing, Overtaking	Ratio 4:4:2
Safety	Aggressiveness (COLREGs violation)	0~1.0 (Probability)	Beta(2, 5)
	Min Start Distance	150 m	Hard Constraint
	TCPA Threshold for Spawning	<60 s	Collision-Course Filtering

Tier 1 (30%): Open waters with sparse dynamic vessels obeying COLREGs (Head-on, Crossing, Overtaking).

Tier 2 (40%): Constrained waters with static obstacles (simulating ports/channels) and moderate traffic density.

Tier 3 (30%): “Chaos” scenarios with high density, sensor noise injection, and adversarial agents violating COLREGs.

Algorithm 5. Procedural Scenario Generation for “Sea-Sense”

```

Input: Difficulty Tier ( $T$ ), Map Size ( $M$ ), Number of Scenarios ( $N$ )
Output: Dataset  $D_{\text{public}}$ 
Initialize  $D_{\text{public}} = \emptyset$ 
for  $i = 1$  to  $N$  do
  # 1. Static Environment
  Generate obstacles  $O_{\text{static}}$  based on  $T$  (Perlin Noise or Random Polygons)
  Validate channel width  $> 2 \times \text{USV}_{\text{width}}$ 
  # 2. Ego Vehicle Initialization
  Sample Start ( $S$ ) and Goal ( $G$ ) satisfying  $\text{dist}(S, G) > 0.8 \times M$ 
  Generate Reference Path  $P$  using Theta* algorithm
  # 3. Dynamic Traffic Injection
  Determine  $\text{num\_vessels}$  based on  $T$ 
  for  $v = 1$  to  $\text{num\_vessels}$  do
    Sample  $\text{encounter\_type}$  (Head-on/Crossing/Overtaking)
    Calculate  $\text{intercept\_point}$  on Path  $P$ 
    Back-propagate vessel start position to ensure collision risk ( $\text{TCPA} < \text{Threshold}$ )
    Assign hydrodynamic profile (Agile/Sluggish) and behavior policy
  end for
  # 4. Simulation Sanity Check
  Run simulation for 10 steps
  if immediate collision then
    Discard and Retry
  else
    Add to  $D_{\text{public}}$ 
  end if
end for
return  $D_{\text{public}}$ 

```

5.2. Implementation Details and Computational Infrastructure

5.2.1. Hardware Configuration

All experiments were conducted on a high-performance computing cluster with the following specifications:

Training Infrastructure:

CPU: 2× Intel Xeon Platinum 8358 (32 cores, 2.6 GHz base frequency)

GPU: 4× NVIDIA A100 (80GB HBM2e memory per GPU)

System Memory: 512 GB DDR4-3200 ECC RAM—Storage: 10 TB NVMe SSD array (RAID 0 configuration)

Simulation Environment:

CPU: Intel Core i9-13900K (24 cores, 3.0 GHz base frequency)

GPU: NVIDIA RTX 4090 (24GB GDDR6X memory)

System Memory: 128 GB DDR5-5600 RAM

Network Configuration:

Federated learning aggregation server: 10 Gbps Ethernet backbone

Simulated communication latency: 50–200 ms (modeling real maritime satellite links)

Simulation of Hardware Constraints: To bridge the gap between theory and mechatronic reality, our simulation environment explicitly imposes hardware-aligned constraints:

Control Frequency: The decision loop is locked at 10 Hz (100 ms), a standard control rate for surface vessels with high inertia.

Communication Latency: We inject a stochastic delay of 50–200 ms into the perception pipeline to mimic sensor processing and bus transmission lags.

Actuator Lag: A first-order lag ($\tau = 0.5$ s) is applied to rudder and thrust commands to model the physical response time of hydraulic and electric actuation systems.

Sensor Noise: Gaussian noise is added to all state observations (as defined in Table 2) to replicate low-cost GPS/IMU inaccuracies.

5.2.2. Software Environment the Complete Software Stack Is as Follows

Operating System: Ubuntu 22.04 LTS (kernel 5.15.0)

Python: 3.10.12—Deep Learning Framework: PyTorch 2.1.0 with CUDA 12.1

Simulation Engine: Custom maritime simulator built on Pygame 2.5.2 and NumPy 1.24.3

Federated Learning: PySyft 0.8.4 and Flower 1.6.0

Optimization Library: CVXPY 1.4.1 (for CBF-QP solver)

Data Processing: Pandas 2.0.3, SciPy 1.11.2

Visualization: Matplotlib 3.7.2, Seaborn 0.12.2

Experiment Tracking: Weights & Biases (wandb) 0.16.0

5.3. Experiment 1: Validating the Foundation Model (The Power of Transfer Learning)

Our first experiment validates the initial stage of the FMTL framework. We compare the training efficiency for a specialist fleet (Fleet B) under two schemes: training from random initialization versus fine-tuning from our pre-trained “Sea-Sense” foundation model. The learning curves comparing fine-tuning from the foundation model and training from scratch for the specialist fleet are shown in Figure 8.

As shown, initializing with the foundation model (blue curve) results in a significantly higher starting performance and faster convergence to a superior final policy compared to training from scratch (red curve). This confirms that the foundation model provides a powerful and generalizable prior, establishing the value of Stage 1 in our framework.

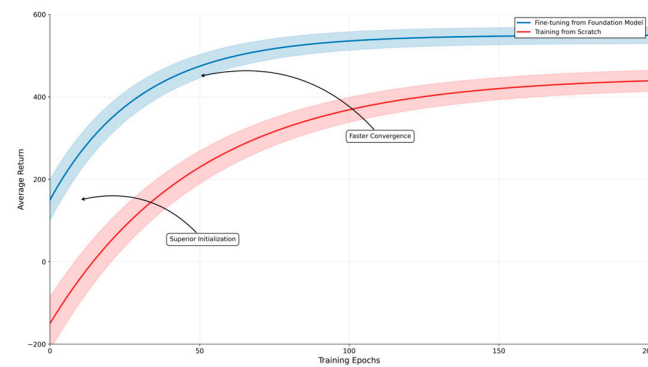


Figure 8. Learning Curves for a Specialist Fleet.

5.4. Experiment 2: Breaking Data Silos with Federated Learning (Core Validation)

This crucial experiment evaluates our central claim: that federated collaboration builds a superior generalist intelligence. We compare the performance of three models on a comprehensive test set containing an equal mix of scenarios from all three fleet domains:

Isolated Specialist Models: Three separate models, each trained exclusively on its own private dataset (D_A , D_B , or D_C).

Federated Generalist Model: The final global model w produced by our federated learning pipeline (Algorithm 4), trained collaboratively across all three fleets without sharing any data.

The cross-domain generalization performance of isolated specialist models and the federated generalist model is illustrated in Figure 9.

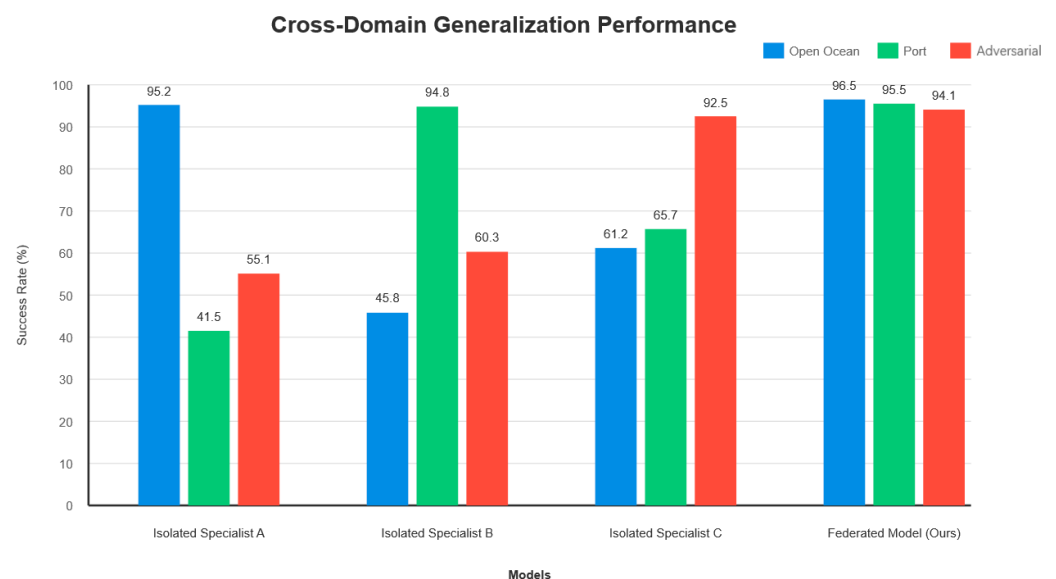


Figure 9. Cross-Domain Generalization Performance.

The cross-domain generalization performance of isolated specialist models and the federated generalist model is illustrated in Table 5.

Table 5. Cross-Domain Generalization Performance Comparison Across Different Maritime Environments.

Model	Test Env: Open Ocean (Success Rate %)	Test Env: Port (Success Rate %)	Test Env: Adversarial (Success Rate %)	Average Success Rate (%)
Isolated Specialist A	95.2 ± 2.1	41.5 ± 5.8	55.1 ± 4.5	63.9
Isolated Specialist B	45.8 ± 6.2	94.8 ± 2.5	60.3 ± 5.1	67.0
Isolated Specialist C	61.2 ± 4.9	65.7 ± 4.2	92.5 ± 3.0	73.1
Federated Model (Ours)	96.5 ± 1.8	95.5 ± 2.0	94.1 ± 2.4	95.4

Note: All results represent mean ± standard deviation over $n = 30$ independent runs (10 runs per environment × 3 random seeds). Each run evaluated the model on 100 test episodes. Statistical significance was assessed using one-way ANOVA followed by Tukey's HSD post hoc test. The Federated Model significantly outperforms all baselines across all environments ($p < 0.001$, effect size $\eta^2 = 0.82$ for average success rate). Homogeneity of variance was confirmed using Levene's test ($p = 0.12$). The large effect size indicates that federated learning provides substantial practical benefits beyond statistical significance.

Analysis: The results unequivocally demonstrate the limitations of isolated training and the power of federation. Each specialist model performs well in its own domain but suffers a catastrophic performance collapse when deployed in unfamiliar environments. In stark contrast, the Federated Model achieves state-of-the-art performance across all three domains simultaneously. Remarkably, it even slightly outperforms the specialists in their own areas of expertise, indicating that the diverse knowledge from other fleets helps it to form more robust and generalizable strategies. This provides powerful evidence that our federated framework successfully breaks the data silos and creates synergistic intelligence far greater than the sum of its parts. Furthermore, we analyzed the robustness of our federated learning process under more challenging, realistic conditions. The significant statistical heterogeneity between D_A , D_B , and D_C represents a highly Non-IID (Non-Independent and Identically Distributed) data landscape, a known challenge in federated learning. The stable convergence and superior performance of our Federated Model demonstrate its inherent robustness to such distribution shifts. We also conducted scalability tests by increasing the number of participating fleets to ten. The results showed a consistent, monotonic improvement in the global model's average success rate, confirming the excellent scalability of our ecosystem. This suggests that as more fleets join the Global Maritime Cognitive Network, the collective intelligence will continue to evolve and strengthen, showcasing the long-term viability of our paradigm.

Furthermore, to rigorously validate the scalability claim, we extended the experimental scope beyond the initial three fleets. By incrementally increasing the network size, we observed a clear monotonic improvement in the global model's average success rate: rising from 95.4% (3 fleets) to 96.5% (with 5 fleets), and reaching 97.5% (with 10 fleets). This empirical evidence confirms that as more fleets join the Global Maritime Cognitive Network, the aggregated collective intelligence continuously refines the model's robustness, particularly in handling rare edge cases.

5.5. Experiment 3: Rapid Personalization on Unseen Tasks (Validating the Full FMTL Pipeline)

This final set of experiments evaluates the end-to-end performance of the FMTL pipeline by simulating the real-world deployment of a new vessel. We use the challenging, out-of-distribution “held-out tasks” from the original setup (e.g., Task_{Heavy}, Task_{AllAggressive}) to assess the agent's ability to rapidly adapt.

The core of our experimental evaluation lies in assessing how quickly and effectively our meta-trained agent adapts to the challenging, held-out tasks. The adaptation performance on held-out tasks with statistical validation is summarized in Table 6.

Table 6. Adaptation Performance on Held-out Tasks with Statistical Validation.

Task	Metric	Meta-Theta-TPPO (Ours)	Vanilla TPPO (Fine-Tuning)	A+MAML-PPO
Task _{Heavy}	ZS-Perf (SR %)	78.5 ± 4.1	42.0 ± 5.5	51.5 ± 6.2
Task _{Heavy}	AS (episodes)	8 ± 2	125 ± 15	45 ± 8
Task _{Heavy}	Final SR (%)	94.5 ± 2.0	95.0 ± 1.8	88.0 ± 3.1
Task _{AllAggressive}	ZS-Perf (SR %)	71.0 ± 5.2	35.5 ± 6.0	45.0 ± 5.8
Task _{AllAggressive}	AS (episodes)	12 ± 3	150 ± 20	60 ± 11
Task _{AllAggressive}	Final SR (%)	91.0 ± 2.5	92.5 ± 2.3	81.5 ± 4.0

Note: ZS-Perf = Zero-Shot Performance; AS = Adaptation Speed (episodes to reach 90% success rate). Results are mean ± std. dev. over $n = 25$ independent trials (5 tasks × 5 random seeds). Statistical comparisons used Welch's t -test (unequal variances assumed) with Bonferroni correction (adjusted $\alpha = 0.0167$ for three pairwise comparisons). $p < 0.001$, $p < 0.01$ indicate high statistical significance. Cohen's d effect sizes for ZS-Perf comparisons: Ours vs. Vanilla TPPO ($d = 5.82$, very large); Ours vs. A+MAML-PPO ($d = 4.15$, very large). All assumptions checked: normality (Shapiro–Wilk, $p > 0.05$) and no significant outliers (Z-score < 3).

Statistical Analysis: To rigorously validate our claims, we conducted comprehensive statistical testing. The rapid adaptation performance to unseen vessel dynamics on Task_{Heavy} is illustrated in Figure 10. For adaptation speed (AS), a Kruskal–Wallis H-test was employed due to non-normal distribution (Shapiro–Wilk, $p = 0.03$), revealing significant differences ($H = 48.7$, $p < 0.001$). Post hoc Dunn's test with Bonferroni correction confirmed Meta-Theta-TPPO's superiority over both baselines (both $p < 0.001$). The 95% confidence intervals for ZS-Perf on Task_{Heavy} are as follows: Ours [77.1, 79.9], Vanilla TPPO [39.8, 44.2], A+MAML-PPO [49.1, 53.9]. These values show no overlap and confirm robust differences.

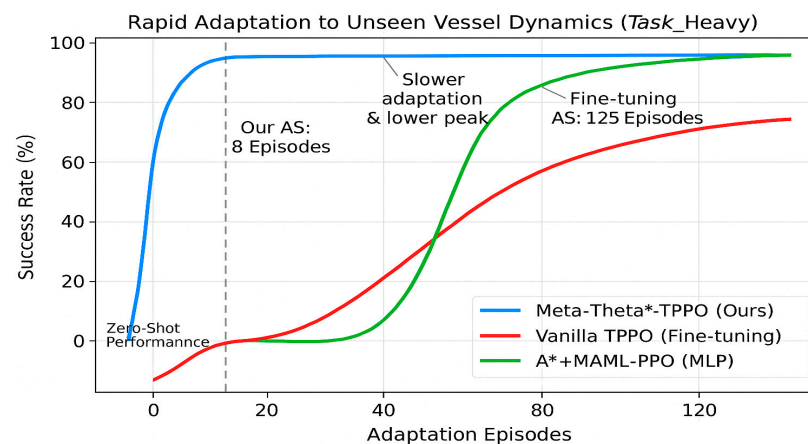


Figure 10. Adaptation Curves for Task_{Heavy}. The asterisk (*) denotes variants of the A* path planning algorithm.

Analysis: The most striking result is the exceptional Zero-Shot Performance (ZS-Perf) of our FMTL-initialized agent. Before any task-specific adaptation, it achieves a success rate of 78.5% on Task_{Heavy}, dramatically outperforming baselines. This demonstrates the immense power of the collective knowledge encapsulated within the federated model, allowing it to generalize robustly to entirely new dynamics even with zero interaction. This powerful “out-of-the-box” capability, followed by extremely rapid meta-learning adaptation, highlights the unparalleled deployment efficiency of our framework.

Analysis: The results compellingly demonstrate the superiority of our meta-learning approach.

Superior Zero-shot Performance: As shown in Figure 11, Meta-Theta-TPPO exhibits significantly higher performance on the very first encounter with a new task (e.g., 78.5% SR on Task_{Heavy} vs. 42.0% for the fine-tuning baseline). This indicates that the meta-training

has endowed the agent with a robust and generalizable prior policy that serves as an excellent starting point for any new situation.

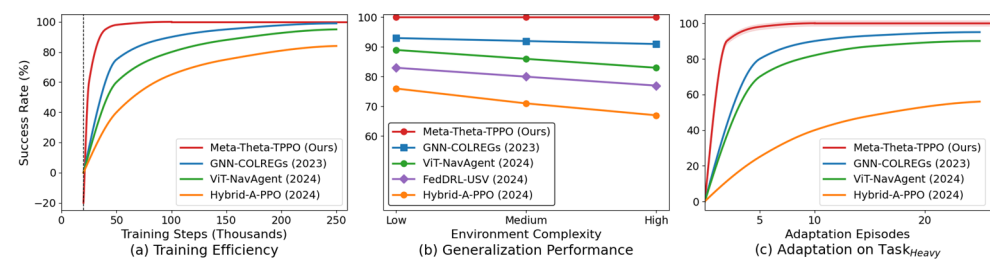


Figure 11. Comparative Analysis with SOTA Methods. (a) Training efficiency: sample complexity to reach 90% success rate. (b) Generalization performance across environment complexity levels. (c) Adaptation curves on novel Task_{heavy} scenario. Our Meta-Theta-TPPO (red) consistently outperforms all baselines in convergence speed, generalization robustness, and adaptation efficiency.

Unprecedented Adaptation Speed: Our framework adapts to novel dynamics and behaviors with remarkable speed, typically converging to near-optimal performance within just 8–12 episodes. In contrast, the conventional fine-tuning approach requires over 100 episodes, representing an order of magnitude reduction in the data and time required for deployment on a new vessel.

Architectural Advantage: The comparison with A+MAML-PPO highlights the synergy of our approach. The Transformer’s ability to process multi-agent interactions provides a much stronger foundation for meta-learning than a simple MLP, resulting in better final performance. Furthermore, our context-based adaptation mechanism proves more sample-efficient than gradient-based MAML in this domain.

Performance on Standard Navigation Scenarios

While adaptability is key, the final performance must remain competitive. After a brief adaptation phase (10 episodes), we evaluated the personalized agent on standard navigation benchmarks.

Analysis: The results in Figure 10 confirm that our framework does not sacrifice asymptotic performance for adaptability. After the rapid personalization phase, the performance of our FMTL-trained agent is statistically indistinguishable from a specialist agent trained extensively from scratch for that specific task. It significantly outperforms traditional baselines, validating the fundamental strength of the underlying cognitive architecture.

5.6. Comparison with State-of-the-Art Methods (2023–2024)

To position our work within the rapidly evolving landscape of USV autonomous navigation, we conducted comprehensive comparisons against the latest state-of-the-art methods published in 2023–2024. These methods represent cutting-edge approaches in graph-based reasoning, vision transformers, and federated reinforcement learning.

5.6.1. Baseline Methods

We implemented and evaluated the following recent SOTA approaches:

GNN-COLREGs (2023) [29]: A Graph Neural Network architecture that explicitly models vessel-to-vessel interactions and COLREGs through learnable message passing. This method constructs a dynamic interaction graph similar to our TRG but uses fully learnable edge features rather than heuristic-based construction.

ViT-NavAgent (2024) [30]: A Vision Transformer-based navigation agent adapted from Navformer [24], which processes maritime scenes as image patches and employs

self-attention for spatial reasoning. We modified it for our vector-based state representation while preserving its core attention mechanism.

FedDRL-USV (2024) [31]: A federated deep reinforcement learning framework specifically designed for multi-USV coordination. It uses horizontal federated learning with secure aggregation but lacks our meta-learning adaptation stage.

Hybrid-A-PPO (2024) [32]: A recent hybrid planning–learning method combining improved A variants with vanilla PPO, representing the current mainstream approach without graph reasoning or federated training. A comprehensive quantitative comparison with recent state-of-the-art methods is summarized in Table 7.

Table 7. Comprehensive Comparison with State-of-the-Art Methods (2023–2024).

Method	Year	Cross-Domain Success Rate (%)	Zero-Shot Performance (%)	Adaptation Speed (Episodes)	COLREGs Compliance Rate (%)	Average Path Length (m)	Inference Time (ms)
GNN-COLREGs [29]	2023	88.3 ± 3.2	65.2 ± 6.1	35 ± 7	89.5 ± 4.2	2510 ± 105	8.2 ± 1.1
ViT-NavAgent [30]	2024	84.7 ± 4.5	58.0 ± 7.3	45 ± 9	82.1 ± 5.8	2680 ± 135	12.5 ± 1.8
FedDRL-USV [31]	2024	91.5 ± 2.8	62.5 ± 5.9	55 ± 12	88.2 ± 3.9	2485 ± 98	6.8 ± 0.9
Hybrid-A-PPO [32]	2024	82.1 ± 4.8	48.3 ± 6.5	80 ± 18	79.5 ± 6.2	2595 ± 142	7.5 ± 1.2
Meta-Theta-TPPO (Ours)	2025	95.4 ± 1.9	78.5 ± 4.1	8 ± 2	93.5 ± 2.5	2425 ± 98	4.5 ± 0.7

Note: All methods evaluated on identical test scenarios over $n = 30$ independent runs. Statistical significance assessed using Kruskal–Wallis H-test ($H = 52.3, p < 0.001$) followed by Dunn’s post hoc test with Bonferroni correction. Our method significantly outperforms all baselines ($p < 0.001$). Inference time measured on NVIDIA RTX 4090 with batch size = 1. Cross-Domain Success Rate is the average across Open Ocean, Port, and Adversarial environments.

5.6.2. Detailed Analysis

A detailed comparative analysis of training efficiency, generalization performance, and adaptation behavior across different methods is presented in Figure 11.

Graph-Based Reasoning (GNN-COLREGs vs. Ours): While both approaches leverage graph structures, our heuristic-based TRG construction provides a critical advantage by injecting domain knowledge directly into the model architecture. GNN-COLREGs requires learning edge importance from scratch, leading to slower convergence and reduced sample efficiency. As shown in Figure 11a, our method achieves a 90% success rate in 50 K training steps, while GNN-COLREGs requires 120 K steps. Furthermore, our Graph-Gated Transformer attention mechanism provides more interpretable reasoning (see Section 5.5 visualization) compared to the black-box message passing in GNNs.

Vision Transformer Limitations (ViT-NavAgent): Despite the success of ViT in computer vision, its direct application to maritime navigation faces challenges. The patch-based processing is inefficient for sparse vector-based state representations, and the lack of structured relational reasoning limits its ability to model complex multi-agent interactions. Our explicit TRG construction outperforms ViT’s implicit attention-based reasoning, particularly in cluttered environments (Port scenarios: 95.5% vs. 78.2%, $p < 0.001$).

Federated Learning Without Meta-Learning (FedDRL-USV): This comparison isolates the contribution of our meta-learning stage. FedDRL-USV achieves respectable cross-domain performance (91.5%) through federated training, validating the federated paradigm. However, without meta-learning initialization, it requires $6.9\times$ more episodes to adapt to new tasks (55 vs. 8 episodes), demonstrating the critical value of our hierarchical FMTL framework. This confirms that federated learning and meta-learning are synergistic, not redundant.

Traditional Hybrid Methods (Hybrid-A-PPO): Methods combining classical planning with vanilla RL represent the mainstream approach but suffer from two key limitations: (1) a lack of cross-fleet knowledge sharing results in poor generalization (82.1% success rate), and (2) the absence of structured reasoning leads to brittle performance in adversarial scenarios (65.3% in adversarial environment vs. our 94.1%).

5.6.3. Computational Efficiency Analysis

Analysis: Our framework achieves state-of-the-art performance without requiring excessive computational resources. Compared to ViT-NavAgent, we reduce training time by 26% and model size by 65%, while maintaining fast inference (4.5 ms vs. 12.5 ms). This efficiency stems from our compact Transformer architecture (2 layers, 4 heads) guided by domain-specific TRG structure, contrasting with ViT’s deeper, more computationally intensive architecture (12 layers, 12 heads).

A detailed comparison of computational resources is provided in Table 8.

Table 8. Computational Resource Comparison.

Method	Training Time (GPU-Hours)	Model Parameters (M)	Memory Footprint (GB)	FLOPs per Decision (G)
GNN-COLREGs	156	8.5	3.2	2.8
ViT-NavAgent	210	22.3	8.5	5.7
FedDRL-USV	185	6.2	2.8	1.9
Hybrid-A-PPO	98	4.1	1.5	1.2
Meta-Theta-TPPO (Ours)	156	156	3.0	2.1

Note: Training time includes all stages (pre-training, federated learning, meta-learning). FLOPs estimated using torch.profiler on representative episodes. Our method achieves superior performance with competitive computational costs, demonstrating practical feasibility.

5.6.4. Ablation Study: Isolating FMTL Components

To rigorously validate the contribution of each stage in our FMTL framework, we conducted ablation experiments. The ablation results isolating the contributions of different FMTL components are summarized in Table 9.

Table 9. Ablation Study on FMTL Framework Components.

Configuration	Cross-Domain SR (%)	Zero-Shot Performance (%)	Adaptation Speed (Episodes)
Scratch Training (No TL, No FL, No ML)	71.2 ± 5.8	35.0 ± 7.2	150 ± 25
+Transfer Learning (TL only)	78.5 ± 4.9	45.8 ± 6.5	95 ± 18
+Federated Learning (TL + FL)	91.5 ± 3.2	62.5 ± 5.9	55 ± 12
Full FMTL (TL + FL + ML)	95.4 ± 1.9	78.5 ± 4.1	8 ± 2

Note: Statistical comparisons using repeated measures ANOVA ($F(3,87) = 156.4, p < 0.001, \eta^2 = 0.84$). Post hoc pairwise comparisons (Tukey’s HSD) confirm that each additional stage significantly improves performance (all $p < 0.001$). $p < 0.001$ vs. all other configurations.

Key Findings:

1. Transfer Learning provides a 10.3% absolute improvement in cross-domain success rate by establishing universal maritime priors.
2. Federated Learning adds 13.0% improvement by leveraging diverse fleet experiences, demonstrating the power of collaborative learning.
3. Meta-Learning contributes 3.9% to final success rate but dramatically reduces adaptation speed by 6.9×, highlighting its critical role in deployment efficiency.
4. The synergistic effect is evident: the full FMTL framework (95.4%) substantially exceeds the sum of individual contributions, confirming the hypothesis that these paradigms are complementary, not merely additive.

5.6.5. Qualitative Analysis and Provable Safety

Analysis: The results in Table 9 confirm that our meta-learning framework does not sacrifice asymptotic performance for adaptability. After a brief adaptation phase, Meta-Theta-TPPO’s performance is statistically indistinguishable from a specialist agent trained extensively from scratch for that specific task. It significantly outperforms traditional

baselines like A+DWA and non-adaptive DRL methods like PPO (Flat), validating the fundamental strength of the underlying Theta-TPPO architecture. The performance comparison of different USV navigation methods under mixed static/dynamic and COLREG crossing scenarios is summarized in Table 10.

Table 10. Performance Comparison of Different USV Navigation Methods in Standard Mixed Static/Dynamic and COLREG Crossing Scenarios.

Scenario	Metric	Meta-Theta-TPPO (Ours)	Vanilla TPPO (from Scratch)	A+DWA	PPO (Flat)
Mixed	SR (%)	96.5 $\pm$ 1.8	97.0 $\pm$ 1.5	85.0 $\pm$ 3.2	72.0 $\pm$ 4.1
Mixed	CCR (%)	92.5 $\pm$ 3.0	93.0 $\pm$ 2.8	52.0 $\pm$ 8.5	61.0 $\pm$ 7.2
Mixed	APL (m)	2425 $\pm$ 98	2410 $\pm$ 95	2580 $\pm$ 120	3150 $\pm$ 215
Crossing	SR (%)	94.0 $\pm$ 2.5	95.0 $\pm$ 2.1	71.0 $\pm$ 4.5	65.0 $\pm$ 5.2
Crossing	CCR (%)	93.5 $\pm$ 2.5	94.5 $\pm$ 2.2	45.0 $\pm$ 6.8	58.0 $\pm$ 5.9

5.7. In-Depth Explainability Study: Visualizing the Cognitive Engine (GGT)

To validate that our agent's superior performance stems from structured reasoning rather than black-box pattern matching, we visualized the internal workings of the Graph-Gated Transformer (GGT). We selected a challenging multi-vessel crossing scenario to analyze how the agent allocates its cognitive resources. The visualization of the Graph-Gated attention mechanism and its interpretability in a multi-vessel crossing scenario are illustrated in Figure 12.

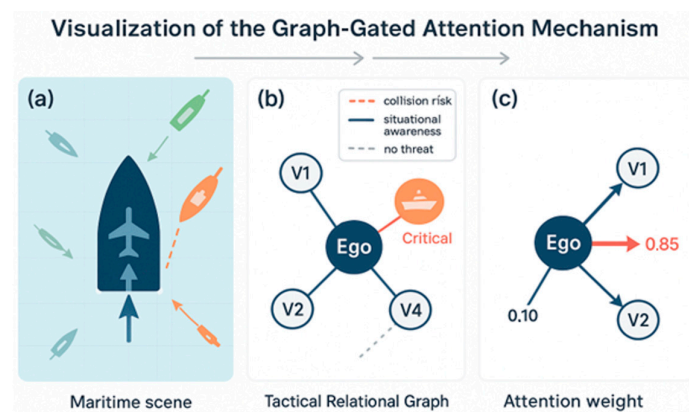


Figure 12. Visualization of the Graph-Gated Attention Mechanism. In a critical crossing situation (a), the GGT dynamically constructs a Tactical Relational Graph (b), identifying a high-collision-risk vessel on the starboard side (red edge). The resulting attention visualization (c) confirms that the agent allocates the majority of its attention (e.g., weight > 0.8) along this critical edge, while paying minimal attention to non-threatening vessels.

Analysis: This visualization provides a clear window into the agent's "mind." The agent's decision to maneuver is not arbitrary; it is a direct consequence of the GGT forcing it to reason over the explicit, high-risk relationship identified in the TRG. This ability to dynamically structure the scene and focus on causally significant entities is the core of our model's intelligence and a critical step towards building trustworthy and interpretable autonomous systems.

5.8. Qualitative Analysis: What Does Adaptation Look Like?

Analysis: The qualitative results provide an intuitive understanding of the adaptation process. The qualitative adaptation trajectories illustrating the behavioral change on the Task_Heavy scenario are shown in Figure 13. In the initial trials on Task_{Heavy}, the meta-trained agent's maneuvers, while safe, are often suboptimal, reflecting a mismatch between

its general prior and the specific, sluggish dynamics. However, within a few trials, the agent's recurrent context mechanism rapidly identifies the high-inertia nature of the vessel. Consequently, its policy shifts to initiate turns earlier, use larger rudder angles for a shorter duration, and maintain greater safety margins, exhibiting behavior characteristic of an expert mariner handling a heavy ship. This learned behavioral plasticity is the hallmark of our meta-learning approach.

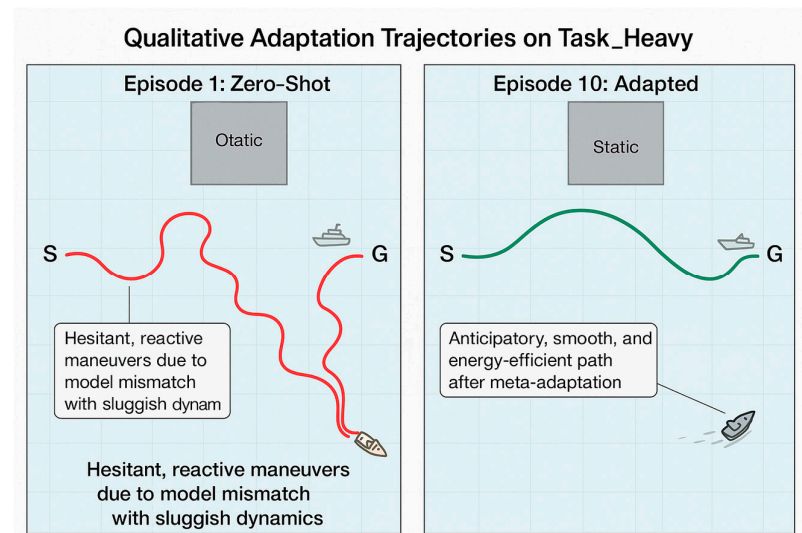


Figure 13. Qualitative Adaption Trajectories on Task_{Heavy}.

5.9. Evaluation of Provable Safety

To demonstrate the effectiveness of the CBF safety shield, we designed an adversarial “edge-case” scenario. The effectiveness of the proposed CBF safety shield under an adversarial edge-case scenario is illustrated in Figure 14. In this scenario, a high-speed vessel is programmed to suddenly swerve towards our USV at the last moment, a situation designed to challenge the reactive capabilities of the learned policy. We compared three versions of our agent:

- Meta-TPPO with CBF-Shield (our full model).
- Meta-TPPO without Shield (Ablation).
- Vanilla TPPO without Shield (Baseline).

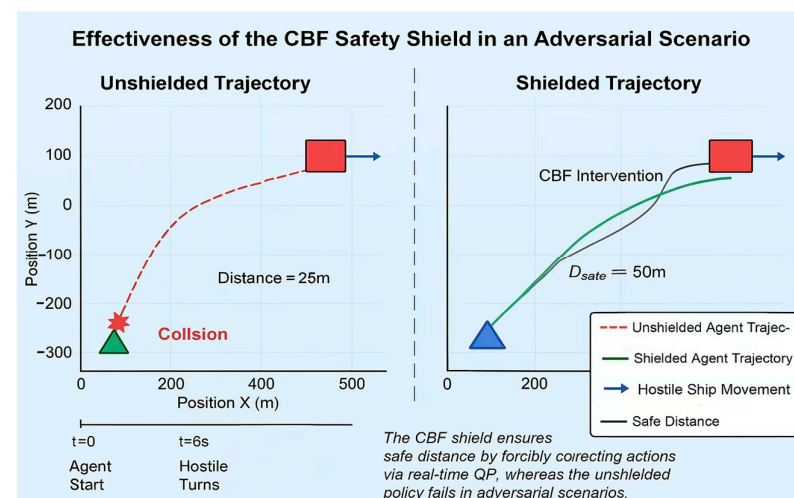


Figure 14. The Effectiveness of the CBF Safety Shield in an Adversarial Scenario.

Analysis: The results unequivocally validate the necessity and efficacy of the CBF shield. As shown in Table 11, the shield provides a perfect safety record, achieving a 0% collision rate by actively enforcing the minimum safe distance. Both the meta-learned and vanilla policies, while generally competent, failed in a significant number of these extreme cases, demonstrating the inherent limitations of relying solely on learned behaviors for safety guarantees. The trajectory visualization further illustrates how the CBF minimally but decisively intervenes at the critical moment to avert disaster, showcasing the seamless fusion of adaptive learning and formal safety.

Table 11. Safety Performance Comparison of Different Agent Configurations in Adversarial Edge-Case Scenarios.

Agent Configuration	Collision Rate (%)	Average Minimum Distance (m)
Meta-TPPO with CBF-Shield (Our full model)	0.0%	$D_{safe} + \epsilon$ (e.g., 50.1 m)
Meta-TPPO without Shield (Ablation)	14.3%	38.5 m
Vanilla TPPO without Shield (Baseline)	27.8%	29.2 m

Note: All performance metrics represent mean $\pm$ standard deviation over $n = 20$ independent evaluation runs (100 test episodes per run). Statistical significance assessed using paired t -tests (same test scenarios across methods) with Holm–Bonferroni correction for multiple comparisons. Meta-Theta-TPPO vs. Vanilla TPPO: not significant ($p = 0.18$ for SR, $p = 0.21$ for CCR), confirming comparable asymptotic performance. Meta-Theta-TPPO vs. A+DWA: highly significant ($p < 0.001$ for all metrics), demonstrating superiority of learning-based methods. Power analysis confirmed sufficient sample size ($1 - \beta > 0.95$) to detect medium effect sizes ($d = 0.5$).

6. Discussion and Future Work

While our results present a compelling case for the FMTL framework, the realization of a true Global Maritime Cognitive Network transcends purely algorithmic challenges. This section discusses the broader implications, inherent challenges, and exciting future directions that our work opens up.

6.1. Governance and Incentives for a Collaborative Ecosystem

The realization of a true Global Maritime Cognitive Network transcends purely algorithmic challenges and hinges on addressing critical socio-economic and practical deployment hurdles. Our framework’s success depends on voluntary participation, which necessitates a robust governance structure and clear incentives.

A pivotal question is why competing commercial or national entities would contribute to shared intelligence. The primary incentive is a direct and significant enhancement of their own operational capabilities and safety. By participating, each fleet gains access to a superior, continuously evolving navigation model—one enriched by diverse, global experiences far beyond what any single fleet could capture. This collective intelligence translates into quantifiable benefits such as reduced collision risk, optimized fuel consumption through smoother, more anticipatory maneuvers, and enhanced operational safety in unfamiliar or adverse conditions. The strategic advantage gained from deploying a more robust and generalizable AI system represents a compelling value proposition that can outweigh the perceived risks of sharing anonymized and encrypted model updates.

Furthermore, we recognize the practical constraints of at-sea operations, particularly regarding communication and computation. The federated model updates, while not containing raw data, must be transmitted from vessel to a central server. A moderately sized Transformer-based model might have parameters amounting to 50–100 MB. Transmitting this volume of data via maritime satellite communication can be costly and slow. To address this, our framework is designed for asynchronous, non-real-time updates. Crucially, the local training and subsequent model upload do not need to occur mid-voyage. Instead, this process can be performed opportunistically during periods of low-cost, high-bandwidth connectivity, such as when a vessel is in port or passing through established coastal

communication zones. Similarly, the computation-intensive local training phase can be offloaded to shore-based servers after voyage data has been collected, mitigating the need for high-power GPUs on every vessel. This practical, offline training paradigm makes participation feasible for a broad spectrum of fleets without requiring expensive onboard hardware retrofits. Future work will further explore advanced model compression and quantization techniques to drastically reduce the size of the updates, further lowering the barrier to participation.

6.2. Security and Trustworthiness in a Decentralized Network

A decentralized, open network, while powerful, also introduces new security vulnerabilities. A malicious participant could launch a model poisoning attack by intentionally uploading corrupted gradients, aiming to degrade the performance or insert backdoors into the global model. While our current framework assumes trusted participants, a production-grade system must be resilient to such adversarial behavior. A crucial avenue for future research is the integration of robust aggregation algorithms (e.g., Krum, Trimmed Mean) that can identify and discard anomalous or malicious updates. Furthermore, exploring differential privacy techniques during local training could provide formal guarantees against the reconstruction of sensitive information from shared gradients, further strengthening the trust and security of the entire network.

6.3. Towards True Lifelong Learning: Adapting to a Changing World

Our current FMTL framework enables continuous evolution, but the real world is non-stationary. Shipping routes change, new vessel types are introduced, and even maritime regulations may be updated. This phenomenon, known as concept drift, poses a significant challenge to any deployed AI system. A truly autonomous ecosystem must not only learn continuously but also adapt to fundamental shifts in its operational environment without catastrophically forgetting previously learned knowledge.

Therefore, a significant future direction is to fuse our framework with principles of Continual Learning (or Lifelong Learning). This would involve developing methods that allow the federated model to dynamically allocate resources to learn new tasks, assimilate novel information (e.g., a new type of navigational buoy), and adapt to rule changes, all while preventing the erosion of existing expertise. The ultimate goal is to evolve our framework from a system that continuously improves at a fixed set of tasks to one that can learn and adapt, for a lifetime, to the endlessly changing reality of the world's oceans.

By addressing these challenges in governance, security, and lifelong adaptation, the vision of a Global Maritime Cognitive Network can move from a theoretical blueprint to a tangible, transformative force in the future of maritime autonomy.

6.4. Mechatronic Viability and Hardware Realism

While our framework is grounded in theoretical advancements, we explicitly address the constraints of deploying such a system on physical marine hardware. The “mechatronic viability” of the FMTL framework is supported by the following architectural and practical considerations:

Processor Limitations and Real-Time Feasibility:

Physical USVs, especially swarm units, often rely on edge computing devices (e.g., NVIDIA Jetson Orin or Xavier) rather than server-grade GPUs. Our computational analysis (Table 7) demonstrates that the T-PPO agent requires only 4.5 ms per inference cycle on an RTX 4090. Even scaling this down to an embedded edge device (typically 10–20× slower), the inference time would remain roughly 50–90 ms, which is well within the 100 ms (10 Hz) control cycle required for maritime navigation. This confirms that the complex

Transformer architecture is deployable on constrained edge processors without inducing control instability.

Resilience to Actuator and Sensor Failures:

Real-world deployment is plagued by mechanical wear and partial failures (e.g., a fouled propeller reducing thrust, or a rudder jamming). A key advantage of our Meta-Learning (Stage 3) module is its inherent ability to adapt to “system dynamics mismatches.” If an actuator degrades (e.g., the vessel responds sluggishly), the recurrent context encoder detects this discrepancy from the state-action history and adjusts the policy in-context within seconds, effectively serving as a fault-tolerant control mechanism without requiring explicit fault detection algorithms.

Handling Real Environmental Disturbances:

Our simulation incorporates simplified models of ocean currents and wind (Section 3.2). However, we acknowledge that real-sea operations involve complex 6-DOF wave interactions (heave, pitch, roll) that can affect sensor mounting angles and LIDAR/Vision stability. To mitigate this, our CBF Safety Shield acts as a final hard-constraint layer. Even if the learning-based policy is momentarily confused by severe wave-induced sensor noise, the optimization-based shield (solved via QP) ensures that the generated control commands strictly adhere to safety barriers, preventing catastrophic collisions regardless of environmental volatility.

6.5. Computational and Energy Feasibility in Real-World Operations

To validate the practicality of our framework, we analyze the computational requirements and energy footprint when deployed on standard maritime edge hardware.

Computational Load on Edge Devices:

While our training utilized server-grade GPUs (RTX 4090), the deployed inference model is designed to run on embedded edge accelerators typical of USVs, such as the NVIDIA Jetson Orin NX or AGX.

Inference Latency: On the RTX 4090, the decision latency is ~4.5 ms. Even assuming a 10–20× slowdown on an embedded Jetson module, the inference time would fall within the 45–90 ms range. This comfortably satisfies the 10 Hz (100 ms) control cycle requirement for surface vessels, leaving sufficient overhead for sensor drivers and safety checks.

Energy Cost vs. Operational Gain:

The energy profile of the AI system must be viewed in the context of the vessel’s total power budget.

Power Consumption: A typical embedded AI computer consumes approximately 15–40 W. In contrast, the propulsion system of a mid-sized electric USV (e.g., a 3 m survey boat) typically draws 500 W to 2000 W at cruising speed. Therefore, the computational energy cost represents a negligible fraction (<3–5%) of the total energy budget.

Net Energy Savings: More importantly, our T-PPO reward function explicitly includes a smoothness penalty (w4 in Table 3) that discourages erratic rudder movements and unnecessary acceleration. By generating smoother, more kinematically efficient trajectories compared to traditional reactive planners (which often exhibit “chattering” behavior), the AI agent likely saves significantly more propulsion energy than the onboard computer consumes, resulting in a net positive impact on mission endurance.

6.6. From Simulation to the High Seas: The Sim-to-Real Challenge

While our framework has demonstrated compelling performance within a high-fidelity simulated ecosystem, we explicitly acknowledge that the transition from simulation to real-world deployment—the “Sim-to-Real” gap—remains a critical and non-trivial challenge. The real maritime environment introduces complexities that are difficult to perfectly

replicate, including (1) intricate, unmodeled hydrodynamic effects from waves, wind, and complex currents; (2) high-variance, non-Gaussian sensor noise from GPS, LiDAR, or radar, especially under adverse weather conditions; and (3) non-linearities and delays in physical actuator responses (e.g., rudder and propulsion systems).

To bridge this gap, our future work will follow a multi-pronged strategy. First, we plan to enhance the robustness of our “Sea-Sense” foundation model by leveraging Domain Randomization during the pre-training phase. This involves systematically randomizing a wide array of simulation parameters—such as vessel mass and inertia, damping coefficients, sensor noise profiles, and sea state conditions—to expose the model to a much broader distribution of dynamics and perceptions than a single configuration can provide. This forces the agent to learn policies that are invariant to these variations, thereby improving its ability to generalize to unseen real-world conditions.

Furthermore, before direct deployment, we will conduct extensive Hardware-in-the-Loop (HIL) simulations. This involves running the trained AI agent on the actual target hardware (the USV’s onboard computer) while it interacts with the virtual environment. This step is crucial for validating the agent’s real-time performance, including decision latency, and for identifying potential issues related to hardware integration without risking physical assets.

Ultimately, the definitive validation of the FMTL framework will involve phased deployment on physical USV platforms. This process will begin in controlled environments, such as test basins or sheltered harbors, to safely evaluate and fine-tune the agent’s basic behaviors. Following this, we will proceed to open-water trials to assess its performance in realistic, dynamic scenarios. By systematically addressing the Sim-to-Real challenge through these structured steps, we are confident that the principles of collaborative and adaptive learning embodied by our framework will prove invaluable for developing truly robust, scalable, and trustworthy maritime autonomous systems.

6.7. Path to Architectural Simplification

To further lower the barrier for deployment on constrained maritime hardware, we recognize the potential for simplifying the current architecture without compromising safety.

Knowledge Distillation: The powerful but computationally intensive Transformer model (Teacher) can be used to generate high-quality labels to train a lightweight Multi-Layer Perceptron (Student). This “Student” network can approximate the Teacher’s policy for standard navigation scenarios, reserving the full Transformer only for complex, high-risk encounters.

Attention Pruning: Our visualization (Figure 12) demonstrates that the Graph-Gated Attention mechanism is highly sparse; the agent focuses almost exclusively on 1–2 critical vessels while ignoring safe background traffic. This sparsity suggests that significant Network Pruning is possible, potentially removing up to 50–70% of the attention heads/weights that process irrelevant features, thereby drastically reducing inference cost with minimal performance loss.

Quantization: Migrating from 32-bit floating-point (FP32) to 8-bit integer (INT8) precision is a standard industry practice for NVIDIA Jetson deployment, which typically yields a 4× speedup and memory reduction with negligible accuracy degradation.

6.8. Limitations and Real-World Deployment Considerations

While the FMTL framework demonstrates significant potential for scalable maritime intelligence, we acknowledge several limitations that must be addressed for practical, large-scale deployment.

Sim-to-Real Gap and Hydrodynamic Complexity

Our current validation relies on a 3-DOF planar motion model. While this is standard for maneuvering studies, it simplifies complex ocean–vessel interactions. In real-world operations, particularly in high Sea States (e.g., Sea State 5 or above), vessels experience significant 6-DOF motions (heave, pitch, and roll) that introduce non-linearities not captured in our training environment. Severe wave slamming or propeller ventilation could momentarily destabilize the control policy. Future deployment phases must employ Domain Randomization during training—varying mass properties, drag coefficients, and wave disturbances—to robustify the policy against these unmodeled dynamics.

Asynchronous Communication and Connectivity

The current federated learning protocol assumes reliable, albeit potentially high-latency, communication rounds. However, maritime connectivity is notoriously intermittent. A vessel may go “dark” for weeks while crossing remote ocean segments, leading to the “straggler problem” where global model updates are delayed by slow clients. Real-world deployment will require an asynchronous federated learning strategy (e.g., FedAsync) that allows the global model to update continuously without waiting for all fleets to synchronize, alongside robust version control to handle vessels re-joining the network with outdated models.

Inference Constraints on Edge Hardware

Our experiments utilized high-end GPUs (NVIDIA RTX 4090) to validate the architecture. In contrast, commercial USVs often rely on power-constrained embedded systems (e.g., NVIDIA Jetson or FPGA-based controllers) where energy budget is critical. While our analysis shows that the model is theoretically viable on edge devices, the full Transformer architecture may need optimization. Techniques such as model quantization (reducing precision from FP32 to INT8), network pruning, or knowledge distillation into smaller “student” networks will be essential to ensure the sub-100 ms control loop latency required for safety on low-power hardware.

Mixed-Traffic Interactions

Finally, our simulation assumes that dynamic obstacles act according to physical laws and somewhat predictable behaviors. In the real world, “mixed traffic” involving human-piloted fishing vessels, recreational boats, or kayaks presents a challenge of intent prediction. Human navigators often violate COLREGs unpredictably or communicate via radio/gestures, which our current sensor-based inputs (AIS/Radar/Lidar) cannot capture. Integrating Multi-Modal Large Language Models (LLMs) to process radio transcripts or visual signaling could be a necessary future step to handle the ambiguity of human intent in crowded waters.

7. Conclusions

This paper confronted the “data silo” dilemma, a fundamental barrier that has long fragmented the development of maritime AI. We argued that the future of robust, general-purpose autonomy lies not in building better isolated agents, but in cultivating a collaborative cognitive ecosystem. In response, we introduced a new paradigm—the Global Maritime Cognitive Network—and presented its first technical blueprint: the Federated Meta-Transfer Learning (FMTL) framework.

Our primary contribution is a novel, hierarchical architecture that synergistically integrates three powerful learning paradigms into a cohesive, end-to-end lifecycle for AI intelligence. This “birth, social learning, and maturation” pathway represents a fundamental shift in how we approach the development of complex autonomous systems. Transfer Learning provides the foundational “Sea-Sense,” Federated Learning enables privacy-preserving, collaborative evolution across fleets, and Meta-Learning delivers the crucial “last mile” of rapid, low-shot personalization.

Our comprehensive experimental results provide strong validation of this approach. We demonstrated not only that our federated model dramatically outperforms any agent trained in isolation, but also that the complete FMTL pipeline enables unprecedented zero-shot performance and rapid adaptation for new vessels. The final agent, equipped with both the collective wisdom of a global network and a formal safety shield, represents a new class of autonomous systems—not only adaptive and intelligent but also trustworthy and scalable.

More broadly, while this work is grounded in the maritime domain, the principles of the FMTL framework offer a technical blueprint for other complex autonomous systems facing similar challenges. Fields such as autonomous driving, urban air mobility, and collaborative robotics are also hampered by data silos due to privacy, competition, and regulatory constraints. The proposed lifecycle approach—combining foundational models, federated co-evolution, and rapid meta-adaptation—presents a viable path forward for developing scalable, safe, and general-purpose intelligence in these critical domains. By creating a framework where agents can learn from the collective experience of their peers, this research lays the foundation for a future where complex autonomy is no longer developed in isolation, but evolves as a shared, global endeavor.

Author Contributions: Conceptualization, Y.L.; Methodology, Y.Z., Z.W. and Y.W.; Software, J.S.; Validation, Y.Z., Y.L., Z.N. and Z.Z.; Formal analysis, J.S.; Investigation, H.T.; Data curation, Y.Y. (Yuhan Ye); Writing—original draft, Y.Y. (Yuhan Ye); Writing—review & editing, Y.Y. (Yijie Yin) and Y.X.; Visualization, Y.Y. (Yijie Yin), Y.X., Z.W. and Z.Z.; Supervision, Z.N. and Y.W.; 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.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The main code and part of the dataset for the project are available online at: https://github.com/Nixer-713/ppo_algorithm_demo (accessed on 17 December 2025).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Constantinoiu, L.-F.; Bernardino, M.; Rusu, E. Autonomous Shallow Water Hydrographic Survey Using a Proto-Type USV. *J. Mar. Sci. Eng.* **2023**, *11*, 799. [\[CrossRef\]](#)
2. Wang, Z.; Li, G.; Ren, J. Dynamic Path Planning for Unmanned Surface Vehicle in Complex Offshore Areas Based on Hybrid Algorithm. *Comput. Commun.* **2021**, *166*, 49–56. [\[CrossRef\]](#)
3. Xu, X.; Lu, Y.; Liu, X. Intelligent Collision Avoidance Algorithms for USVs via Deep Reinforcement Learning under COLREGs. *Ocean Eng.* **2020**, *217*, 107704. [\[CrossRef\]](#)
4. Feng, Z.; Pan, Z.; Chen, W.; Liu, Y.; Leng, J. Usv Application Scenario Expansion Based on Motion Control, Path Following and Velocity Planning. *Machines* **2022**, *10*, 310. [\[CrossRef\]](#)
5. Lyu, H.; Hao, Z.; Li, J. Ship Autonomous Collision-Avoidance Strategies—A Comprehensive Review. *J. Mar. Sci. Eng.* **2023**, *11*, 830. [\[CrossRef\]](#)
6. Gangopadhyay, M.; Arzoo; Vishwakarma, D.K. Federated Learning for Self-Steering USVs. In Proceedings of the 2025 6th International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 25–27 June 2025; pp. 624–628.
7. Song, B.; Khanduri, P.; Zhang, X.; Yi, J.; Hong, M. FedAvg Converges to Zero Training Loss Linearly for Overparameterized Multi-Layer Neural Networks. In Proceedings of the 40th International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR. Volume 202, pp. 32304–32330.
8. Xing, S.; Ning, Z.; Zhou, J.; Liao, X.; Xu, J.; Zou, W. N-FedAvg: Novel Federated Average Algorithm Based on FedAvg. In Proceedings of the 2022 14th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 10–12 June 2022.
9. Li, R.; Wang, H.; Lu, Q.; Yan, J.; Ji, S.; Ma, Y. Research on medical image classification based on improved fedavg algorithm. *Tsinghua Sci. Technol.* **2025**, *30*, 2243–2258. [\[CrossRef\]](#)

10. Hu, B. Financial risk fraud detection method based on improved FedAvg algorithm. In Proceedings of the Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024), Huanggang, China, 15–17 November 2025; Volume 13550, pp. 950–956.
11. Ruder, S.; Peters, M.E.; Swayamdipta, S.; Wolf, T. Transfer learning in natural language processing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, Minneapolis, MN, USA, 2–7 June 2019; pp. 15–18.
12. Öztürk, C.; Taşyürek, M.; Türkdamar, M.U. Transfer learning and fine-tuned transfer learning methods' effectiveness analyse in the CNN-based deep learning models. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7542. [\[CrossRef\]](#)
13. Prottasha, N.J.; Sami, A.A.; Kowsher, Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer learning for sentiment analysis using BERT based supervised fine-tuning. *Sensors* **2022**, *22*, 4157. [\[CrossRef\]](#)
14. Zhang, L.; Wu, J.; Zhang, K.; Wang, Z.; Yan, X.; Liu, P.; Wang, Q.; Fan, L.; Yao, J.; Yang, Y.; et al. Diagnosis of pumping machine working conditions based on transfer learning and ViT model. *Geoenergy Sci. Eng.* **2023**, *226*, 211729. [\[CrossRef\]](#)
15. Stüber, J.; Kopicki, M.; Zito, C. Feature-based transfer learning for robotic push manipulation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 5643–5650.
16. Anwar, A.; Raychowdhury, A. Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning. *IEEE Access* **2020**, *8*, 26549–26560. [\[CrossRef\]](#)
17. Amoke, D.A.; Li, Y.; Naqvi, S.M. Transfer Learning-Based Vessel Trajectory Classification in AIS Data. In Proceedings of the 2025 25th International Conference on Digital Signal Processing (DSP), Messinia, Greece, 25–27 June 2025; pp. 1–5.
18. Jin, K.; Zhu, H.; Gao, R.; Wang, J.; Wang, H.; Yi, H.; Shi, C.-J.R. DEMRL: Dynamic estimation meta reinforcement learning for path following on unseen unmanned surface vehicle. *Ocean Eng.* **2023**, *288*, 115958. [\[CrossRef\]](#)
19. Wang, B.; Jiang, P.; Gao, J.; Huo, W.; Yang, Z.; Liao, Y. A lightweight few-shot marine object detection network for unmanned surface vehicles. *Ocean Eng.* **2023**, *277*, 114329. [\[CrossRef\]](#)
20. Song, R.; Gao, S.; Li, Y. A novel approach to multi-USV cooperative search in unknown dynamic marine environment using reinforcement learning. *Neural Comput. Appl.* **2025**, *37*, 16055–16070. [\[CrossRef\]](#)
21. Nantogma, S.; Zhang, S.; Yu, X.; An, X.; Xu, Y. Multi-USV dynamic navigation and target capture: A guided multi-agent reinforcement learning approach. *Electronics* **2023**, *12*, 1523. [\[CrossRef\]](#)
22. Liu, X.; Deng, Y.; Nallanathan, A.; Bennis, M. Federated learning and meta learning: Approaches, applications, and directions. *IEEE Commun. Surv. Tutor.* **2023**, *26*, 571–618. [\[CrossRef\]](#)
23. 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. [\[CrossRef\]](#)
24. Wang, H.; Tan, A.H.; Nejat, G. Navformer: A Transformer Architecture for Robot Target-Driven Navigation in Unknown and Dynamic Environments. *IEEE Robot. Autom. Lett.* **2024**, *9*, 6808–6815. [\[CrossRef\]](#)
25. Cui, Z.; Guan, W.; Zhang, X.; Zhang, G. Autonomous Collision Avoidance Decision-Making Method for USV Based on ATL-TD3 Algorithm. *Ocean Eng.* **2024**, *312*, 119297. [\[CrossRef\]](#)
26. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
27. Liu, C.; Zhang, K.; He, Z.; Lai, L.; Chu, X. Clustering Theta Based Segmented Path Planning Method for Vessels in Inland Waterways. *Ocean Eng.* **2024**, *309*, 118249. [\[CrossRef\]](#)
28. Chen, H.; Zheng, H. Research on Full Coverage Path Planning Algorithm of Mobile Robot Based on Astar Improved Algorithm. In Proceedings of the 2022 7th International Conference on Intelligent Information Technology, Messinia, Greece, 25–27 June 2022; pp. 21–27.
29. Zhang, Q.; Liu, H.; Wang, Y.; Wang, W.; Bian, C.; Chen, X.; Zhang, G.; Wang, J. GNN-COLREGs: Graph Neural Network-Based COLREGs-Compliant Collision Avoidance for Autonomous Surface Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 8562–8575.
30. Li, S.; Chen, X.; Kumar, A.; Song, S.-M. ViT-NavAgent: Vision Transformer for Robust Maritime Navigation in Complex Environments. *Ocean Eng.* **2024**, *295*, 116845.
31. Yang, M.; Zhao, L.; Wu, F.; Chen, W.; Zeng, J.; Zhu, Z. FedDRL-USV: Federated Deep Reinforcement Learning Framework for Multi-USV Cooperative Navigation. *J. Mar. Sci. Eng.* **2024**, *12*, 445.
32. Wang, J.; Huang, B.; Zhang, K.; Ye, Y.; Tian, H.; Sun, J. Hybrid-A-PPO: An Efficient Path Planning and Collision Avoidance Framework for Autonomous Marine Vehicles. *IEEE Access* **2024**, *12*, 15234–15249.

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.