CEG5003

Physical-informed Neural Networks AI model for Carbon Emission Estimation

Name: HAN YUCHEN
ID: A0298841M
Department of Electrical Computer Engineering, NUS

NUS
National University
of Singapore

April 14, 2025

Contents

Abstract

High-precision modeling of ship motion and carbon emissions is critical for intelligent shipping and low-carbon transformation. However, the inherent nonlinearities and multi-variable couplings in ship propulsion systems pose significant challenges to both purely physics-based and data-driven approaches, often resulting in limited prediction accuracy and poor noise robustness. To address these limitations, this paper proposes a physics-informed neural network (PINN) framework as a gray-box solution that integrates both white-box and black-box modeling paradigms. The proposed hybrid architecture combines the MMG (Maneuvering Modeling Group) model to provide physically consistent constraints, and a residual attention neural network trained on observational data to capture complex nonlinear dynamics. A composite loss function—consisting of data loss and physics-based loss derived from MMG dynamics—is adaptively weighted during training and backpropagated to optimize the network parameters. Experimental results on the SIMMAN2008 free-running dataset demonstrate the effectiveness of the proposed approach in predicting both ship motion states and specific fuel oil consumption (SFOC), with SFOC prediction errors consistently maintained within the range of 0.18%–0.39%. The model also exhibits strong robustness under noise perturbations, offering a reliable, stable, and interpretable solution for full-process ship emission estimation in support of maritime decarbonization.

Keywords :Ship maneuvering, physics-informed, Hybrid modeling, Specific fuel oil consumption (SFOC), Neural network

1 Introduction

The maritime transportation sector is responsible for approximately 2.9% of global greenhouse gas (GHG) emissions, making it a critical focus in the global push toward carbon neutrality [1]. In response, regulatory bodies such as the International Maritime Organization (IMO) have introduced energy efficiency and carbon intensity metrics, including the Energy Efficiency Existing Ship Index (EEXI) and the Carbon Intensity Indicator (CII), to enforce low-emission ship operations [2, 3]. This shift has created a growing demand for accurate and interpretable models that can predict fuel consumption and emissions from ship motion and control inputs, in support of intelligent route planning, energy optimization, and digital twin systems [4, 5].

Ship maneuvering modeling serves as the foundation for designing simulators and control systems [6, 7]. Mechanism-based models—also known as parametric models—such as the MMG (Maneuvering Modeling Group) model, establish differential equations based on physical principles, incorporating hydrodynamic, rudder, and propeller forces to capture three-degree-of-freedom motion [8]. These white-box models are valued for their physical interpretability and have been widely adopted in academia and industry. However, they require extensive parameter identification using captive model tests, computational fluid dynamics (CFD), or system identification from sea trials [9, 10]. Additionally, their accuracy is limited by incomplete physical knowledge and heavy reliance on expert tuning, making them less adaptable to varied hull forms or environmental conditions.

To overcome these limitations, data-driven or nonparametric methods have emerged as a promising alternative. These approaches learn input–output mappings without explicitly modeling physical dynamics, offering fast deployment and high flexibility [11]. In particular, neural networks have gained popularity as powerful surrogate models due to their strong nonlinear approximation capability and robustness to noise [12, 13]. Previous studies have applied Long Short-Term Memory (LSTM) network, Chebyshev basis networks, symbolic regression and hybrid random forest approaches to model ship dynamics [**symbolic Random_forrest**, 14, 15]. However, such methods often suffer from high model complexity, manual architecture design, and limited transferability across vessel types. Recent research suggests that shallow networks can achieve competitive performance for maneuvering modeling while maintaining computational simplicity [16].

To combine the interpretability of physical models with the adaptability of learning systems, hybrid or gray-box models have attracted growing interest. Physics-Informed Neural Networks (PINNs) represent a notable development in this direction, embedding physical laws into the training process via customized loss functions [17, 18]. PINNs have demonstrated promise in robust system identification and noise-tolerant learning, but many applications remain confined to isolated subsystems or idealized simulations [19, 20].

A holistic, end-to-end framework that connects ship dynamics, propulsion behavior, and carbon emissions in a physically consistent manner remains largely underexplored.

In this work, we propose a hybrid learning framework that unifies white-box, black-box, and gray-box modeling paradigms into a cohesive structure. The MMG model provides interpretable baseline predictions based on physical laws; a residual attention network corrects the modeling error using learned data patterns; and a physics-informed neural network integrates both through adaptive loss weighting. The pipeline is extended to include a lightweight multilayer perceptron (MLP) that estimates Specific Fuel Oil Consumption (SFOC) from propulsion-derived variables. Experiments conducted on the SIMMAN2008 free-running dataset demonstrate that the proposed framework achieves accurate and robust prediction under noisy conditions, offering a viable solution for intelligent and low-carbon maritime operations.

2 Methodology

To model the coupled dynamics of ship motion and carbon emissions, we propose a two-stage hybrid architecture that integrates physical principles and neural networks in a unified end-to-end pipeline, as illustrated in Fig. 1.

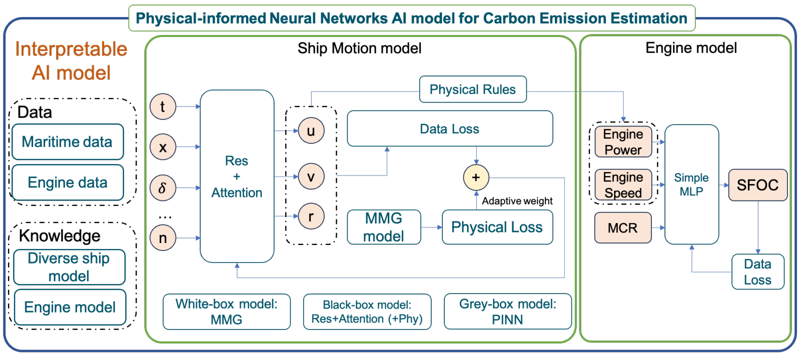


Figure 1: Hybrid Modeling Framework

In the first stage, ship motion is predicted based on maritime control and state data. The black-box module employs a residual attention neural network to construct a data-driven prediction model for the ship's motion states. The network is trained to minimize the discrepancy between its outputs and the observed states, forming the basis of the data loss component.

In parallel, the MMG model—used as the white-box reference which provides physically grounded motion dynamics using hydrodynamic equations—computes the physical accelerations based on current states and control inputs, and predicts the next-step motion using numerical integration. The difference between the MMG-based next-step states and the real next-step states is formulated as the physical loss. This term penalizes violations of physical consistency and acts as a regularization constraint to stabilize training, forming the core of the gray-box PINN architecture.

In the second stage, the predicted motion states are used to derive propulsion variables such as engine power and shaft speed. These variables are passed into a simple MLP model, which estimates the ship's Specific Fuel Oil Consumption (SFOC). The emission prediction module is trained using standard mean squared error loss.

This modular system architecture integrates the interpretability of physical modeling, the nonlinear fitting capability of data-driven methods, and the scalability of emission estimation. It exhibits strong generalization and transferability, while maintaining stable performance even under noisy conditions.

2.1 Ship Motion Modeling

2.1.1 White-box: MMG-based Physical Model

The white-box component of our hybrid framework is built upon the Maneuvering Modeling Group (MMG) model[8], a widely used semi-empirical approach for predicting ship maneuvering behavior. Originally developed in Japan in the 1970s, the MMG model has become the de facto standard for simulating free-running tests, sea trials, and IMO-compliant maneuvering performance evaluations. It is particularly effective in capturing the coupled dynamics between hull hydrodynamics, rudder effects, and propeller thrust under low-speed and large-amplitude maneuvering conditions.

In the MMG model, the ship is treated as a rigid body moving in the horizontal plane. The 3-DOF (degrees of freedom) motion states include surge velocity u, sway velocity v, and yaw angular velocity r. The time evolution of these states is governed by Newton–Euler equations, with external forces decomposed into three components: hull forces, rudder forces, and propeller thrust:

$$\left.\begin{aligned} (m+m_x)\dot{u} - (m+m_y)v_m - x_G m r^2 &= X_H + X_P + X_R \\ (m+m_y)\dot{v}_m + (m+m_x)ur + x_G m\dot{r} &= Y_H + Y_R \\ (I_{zG} + x_G^2 + J_z)\dot{r} + x_G m(\dot{v}_m + ur) &= N_H + N_R \end{aligned}\right\} \tag{1}$$

where m is the ship mass, I_z is the moment of inertia about the vertical axis, and m_x, m_y, J_z are added mass and added moment of inertia terms, x_G is longitudinal coordinate of center of gravity of ship. Subscript H, R, and P means hull, rudder, and propeller, respectively.

(1) Hull Forces. The hydrodynamic forces X_H; Y_H and N_H generated by the hull are expressed as follows::

$$\left.\begin{aligned} X_H &= (1/2)\rho L_{pp} d U^2 X'_H(v'_m, r') \\ Y_H &= (1/2)\rho L_{pp} d U^2 Y'_H(v'_m, r') \\ N_H &= (1/2)\rho L_{pp} d U^2 N'_H(v'_m, r') \end{aligned}\right\} \tag{2}$$

where v'_m denotes non-dimensionalized lateral velocity defined by v_m/U, which $U = \sqrt{u^2 + v_m^2}$. r' denotes non-dimensionalized yaw rate defined by rL_{pp}/U. X'_H is expressed as the sum of resistance coefficient R_0 and the 2nd and 4th order polynomial function of v'_m, r'. Y'_H, N'_H are expressed as the 1st and 3rd order polynomial function of v'_m, r':

$$\left.\begin{aligned} X'_H(v'_m, r') &= -R_0 + X_{vv} v'^2_m + X_{vr} v'_m r' + X_{rr} r'^2 + X_{vvvv} v'^4_m \\ Y'_H(v'_m, r') &= Y_v v'_m + Y_R r' + Y_{vvv} v'^3_m + Y_{vvr} v'^2_m r' + Y_{vrr} v'_m r'^2 + Y_{rrr} r'^3 \\ N'_H(v'_m, r') &= N_v v'_m + N_R r' + N_{vvv} v'^3_m + N_{vvr} v'^2_m r' + N_{vrr} v'_m r'^2 + N_{rrr} r'^3 \end{aligned}\right\} \tag{3}$$

where $R_0, X_{vv}, X_{vr}, X_{rr}, X_{vvvv}, Y_v, Y_R, Y_{vvv}, Y_{vvr}, Y_{vrr}, Y_{rrr}, N_v, N_R, N_{vvv}, N_{vvr}, N_{vrr}, N_{rrr}$ are called the hydrodynamic derivatives on maneuvering, which are generally dimensionless and obtained through captive model tests or validated empirical values (e.g., KVLCC2 datasets). .

(2) Propeller Thrust. Surge force due to propeller X_P is expressed as

$$X_P = (1 - t_P) \cdot T \tag{4}$$

Thrust deduction factor t_P is assumed to be constant at given propeller load for simplicity. Propeller thrust T is written as

$$\begin{aligned} T &= \rho n_p^2 D_p^4 K_T(J_p) \\ J_p &= \frac{u(1 - w_p)}{n_p D_p} \end{aligned} \tag{5}$$

Where n_p is propeller revolution, D_p is the propeller diameter, ρ is the water density. K_T is approximately expressed as 2nd polynomial function of propeller advanced ratio J_p:

$$K_T(J_p) = k_2 J_p^2 + k_1 J_p + k_0 \tag{6}$$

The w_p represent wake coefficient at propeller position in maneuvering motions, which is expressed as:

$$(1 - w_p)/(1 - w_{p0}) = 1 + (1 - exp(-C_1|\beta_p|))(C_2 - 1) \tag{7}$$

Where w_{p0} is wake coefficient at propeller position in straight moving β_p is the geometrical inflow angle to the propeller in maneuvering motions.

(3) Rudder Forces. The rudder is treated as a lifting surface. The rudder-induced forces and moments are:

$$\left.\begin{aligned} X_R &= -(1 - t_R)F_N \sin(\delta) \\ Y_R &= -(1 + a_H)F_N \cos(\delta) \\ N_R &= -(x_R + a_H x_H)F_N \cos(\delta) \end{aligned}\right\} \tag{8}$$

where F_N is the rudder normal force, t_R, a_H, x_H are the coefficients representing mainly hydrodynamic interaction between ship hull and rudder.

The parameter t_R represents the effective reduction in axial force generated by the rudder due to hull resistance and interaction effects. It modifies the rudder's longitudinal force component $F_N \sin\delta$. A typical value for t_R ranges from 0.3 to 0.5 depending on hull form and propeller-rudder configuration. It accounts for the combined effect of rudder drag and increased propeller thrust in off-axis flows.

The coefficient a_H reflects the increase in lateral force acting on the hull induced by rudder deflection. Empirical values of a_H typically lie in the range 0.3–0.4, suggesting a 30–40% increase in hull-side force due to rudder activity.

The parameter x_H represents the effective longitudinal location (moment arm) where the additional lateral force (induced by the rudder) acts on the hull. The value of x_H is usually negative (aft of midship), commonly taken as $x_H \approx -0.45L_{pp}$, reflecting that the interaction force acts near the stern area.

Rudder normal force F_N is expressed as:

$$F_N = (1/2)\rho A_R U_R^2 f_\alpha \sin(\alpha_R) \tag{9}$$

Here, the resultant rudder inflow velocityU_R and the angle a_R are expressed as

$$\begin{aligned} U_R &= \sqrt{u_R^2 + v_R^2} \\ \alpha_R &= \delta - tan^{-1}(\frac{v_R}{u_R}) \approx \delta - \frac{v_R}{u_R} \end{aligned} \tag{10}$$

Where A_R is profile area of movable part of mariner rudder, f_α is rudder lift gradient coefficient, reflecting the linear relation between lift force and angle of attack under small-angle assumptions. Both paras are constant.

(4) KVLCC2 Hull Form and Model Parameters. To support the MMG modeling and validation, this study adopts the KVLCC2 hull form as the representative vessel. KVLCC2 is a Very Large Crude Carrier (VLCC) model developed by the Korea Research Institute of Ships and Ocean Engineering (KRISO), and it has been widely used in benchmark studies such as SIMMAN2008 for maneuvering prediction and CFD validation.

In this work, both free-running test data and hydrodynamic derivatives for KVLCC2 are utilized. The principal particulars of the full-scale ship are listed below:

Table 1: Principal particulars of a KVLCC2 tanker

	L3-model	**L7-model**	**Fullscale**
Scale	1/110	1/45.7	1.00
L_{pp} (m)	2.902	7.00	320.0
B (m)	0.527	1.27	58.0
d (m)	0.189	0.46	20.8
∇ (m^3)	0.235	3.27	312,600
x_G (m)	0.102	0.25	11.2
C_b	0.810	0.810	0.810
D_P (m)	0.090	0.216	9.86
H_R (m)	0.144	0.345	15.80
A_R (m^2)	0.00928	0.0539	112.5

Table 2: Resistance coefficient and hydrodynamic derivatives on maneuvering

X-direction		Y-direction		N-direction	
R_0	0.022	Y_v	−0.315	N_v	−0.137
X_{vv}	−0.040	Y_R	0.083	N_R	−0.049
X_{vr}	0.002	Y_{vvv}	−1.607	N_{vvv}	−0.030
X_{rr}	0.011	Y_{vvr}	0.379	N_{vvr}	−0.294
X_{vvvv}	0.771	Y_{vrr}	−0.391	N_{vrr}	0.055
		Y_{rrr}	0.008	N_{rrr}	−0.013

The model used in simulations includes verified added mass coefficients and empirical hydrodynamic derivatives from published literature and SIMMAN2008 test data. All simulations are carried out under deep-water conditions with a nominal propeller speed of 614 RPM, and the sampling interval is 0.05s.

2.1.2 Black-box: Pure Data-Driven ResidualAttentionNN Architecture

To directly model the mapping from current navigation and control states to ship motion dynamics, we propose a deep residual neural network architecture augmented with multi-head self-attention, referred to as **ResidualAttentionNN**, as illustrated in Fig. 2. This model captures complex nonlinear relationships without relying on any physical prior, serving as a baseline black-box predictor.

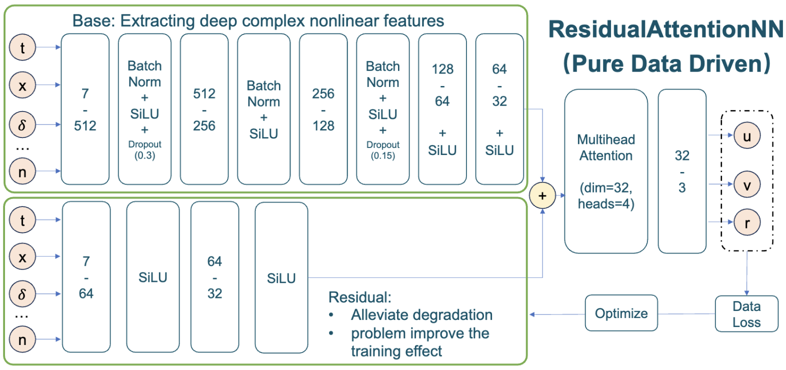


Figure 2: Pure Data-Driven ResidualAttentionNN

The overall architecture is composed of the following components:

(1) Input Embedding Layer. The input vector consists of 7 variables: time t, position (x, y), rudder angle δ, propeller speed n_p, and optionally heading angle ψ and roll angle p. These are linearly projected into a 512-dimensional latent space:

$$\mathbf{h}_0 \in \mathbb{R}^{512} \leftarrow \text{Linear}(7 \rightarrow 512)$$

(2) Deep Feature Extractor (Main Tower). The main branch consists of six fully connected layers with progressively reduced dimensions:

- $512 \rightarrow 256$ (BatchNorm + SiLU + Dropout(0.3))
- $256 \rightarrow 256$ (BatchNorm + SiLU)
- $256 \rightarrow 128$ (BatchNorm + SiLU + Dropout(0.15))
- $128 \rightarrow 64$ (SiLU)
- $64 \rightarrow 32$ (SiLU)

(3) Residual Shortcut Branch. A parallel lightweight branch processes the raw input through:

$$\text{Linear}(7 \rightarrow 64 \rightarrow 32)$$

Both layers use SiLU activation. The result is added to the main branch output to form the residual connection:

$$\mathbf{h}_{\text{sum}} = \mathbf{h}_{\text{main}} + \mathbf{h}_{\text{res}}$$

(4) Multi-Head Attention Layer. A multi-head self-attention module (4 heads, dimension 32) processes the fused representation:

$$\text{MultiHead}(Q, K, V) \in \mathbb{R}^{32}$$

(5) Output Layer. The final output is a linear projection:

$$[\hat{u}, \hat{v}, \hat{r}] = \text{Linear}(32 \to 3)$$

(6) Loss Function. To ensure robustness against noisy measurements and improve stability during training, we adopt the Smooth L1 loss (also known as the Huber loss) as the primary regression criterion. Unlike the standard L_2 loss which is sensitive to outliers, the Smooth L1 loss behaves quadratically for small errors and linearly for large errors. It is defined as:

$$\mathscr{L}_{\text{SmoothL1}}(x, y) = \begin{cases} \frac{1}{2}(x-y)^2, & \text{if } |x-y| < \beta \\ \beta \cdot (|x-y| - \frac{1}{2}\beta), & \text{otherwise} \end{cases} \tag{11}$$

where β is a transition threshold, typically set to $\beta = 1$ in practice. The total loss for motion state prediction is computed as:

$$\mathscr{L}_{\text{data}} = \text{SmoothL1Loss}(\hat{\mathbf{y}}, \mathbf{y}^{\text{true}}) = \text{SmoothL1Loss}([\hat{u}, \hat{v}, \hat{r}], [u, v, r]) \tag{12}$$

This loss function balances precision and robustness, making it particularly suitable for real-world trajectory data with measurement noise.

This pure data-driven model is used as a performance reference in our hybrid learning framework. While it offers strong expressive capacity, its lack of physical constraints motivates the introduction of residual correction and physics-informed regularization in subsequent sections.

2.1.3 Black-box: ResidualAttentionNN with Physics Rules Correction

To leverage the interpretability of physics-based modeling while retaining the flexibility of neural networks, we implement a residual learning architecture that explicitly corrects the output of the MMG physical model, as illustrated in Fig. 3. This hybrid formulation preserves the structure and constraints of ship dynamics while compensating for modeling inaccuracies via data-driven residuals.

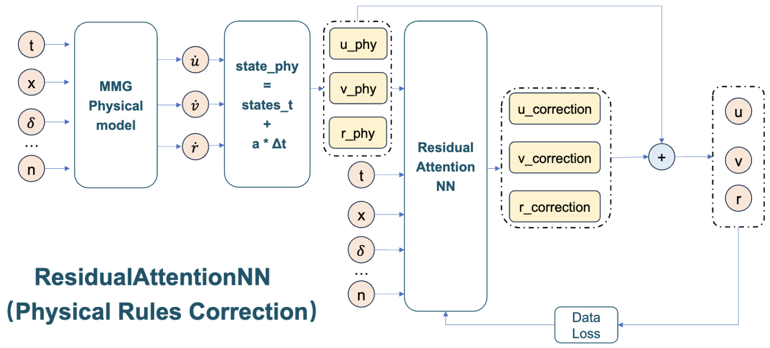


Figure 3: ResidualAttentionNN with Physics Rules Correction

The overall process consists of the following steps:

(1) Physical Model Prediction. Given the current navigation states and control inputs $(t, x, \delta, n, \dots)$, the MMG physical model is used to compute the dynamic accelerations $(\dot{u}, \dot{v}, \dot{r})$ through 3-DOF equations. These accelerations are integrated over a time step Δt to obtain physically predicted motion states:

$$[u_{\text{phy}}, v_{\text{phy}}, r_{\text{phy}}] = [u_t, v_t, r_t] + [\dot{u}, \dot{v}, \dot{r}]_{\text{MMG}} \cdot \Delta t$$

(2) Residual Learning Network. The predicted physical state $\mathbf{u}_{\text{phy}}$ is concatenated with current control features $(t, x, \delta, n, \dots)$ and fed into a ResidualAttentionNN module. The network is designed to learn corrective residuals $\Delta\mathbf{u} = (\Delta u, \Delta v, \Delta r)$ for each of the motion components.

The final prediction is given by:

$$[\hat{u}, \hat{v}, \hat{r}] = [u_{\text{phy}}, v_{\text{phy}}, r_{\text{phy}}] + [\Delta u, \Delta v, \Delta r]$$

(3) Architecture Design. The ResidualAttentionNN structure follows the same backbone as the pure data-driven model (see previous section), including:

- Input projection of $(t, x, ..., \delta, n, u_{\text{phy}}, v_{\text{phy}}, r_{\text{phy}})$ to latent space
- Multi-layer residual blocks with BatchNorm, SiLU, and dropout
- Multi-head self-attention module for interaction modeling
- Output layer projecting to $(\Delta u, \Delta v, \Delta r)$

(4) Loss Function. Only the corrected outputs $\hat{u}, \hat{v}, \hat{r}$ are compared against ground truth using a regression loss. We use the Smooth L1 loss function for robustness:

$$\mathcal{L}_{\text{data}} = \text{SmoothL1Loss}(\hat{\mathbf{y}}, \mathbf{y}^{\text{true}}) = \text{SmoothL1Loss}([\hat{u}, \hat{v}, \hat{r}], [u, v, r]) \tag{13}$$

The residual correction framework effectively combines the interpretability of physics-based models with the expressiveness of data-driven learning. By anchoring the prediction to the MMG physical model and allowing the neural network to learn only the correction term, the network focuses on compensating for systematic modeling errors—such as unmodeled dynamics, parameter uncertainties, or scale mismatches—rather than re-learning the entire physical process. This not only reduces the learning burden and training data requirement but also improves model stability, and physical plausibility.

2.1.4 Grey-box: Physics-Informed Neural Network (PINN)

To further enhance the physical consistency and generalization capability of the model, we integrate the ResidualAttentionNN with MMG physical model via a physics-informed training strategy. This leads to a grey-box modeling structure, known as the Physics-Informed Neural Network (PINN), which jointly optimizes both data fidelity and physical law adherence.

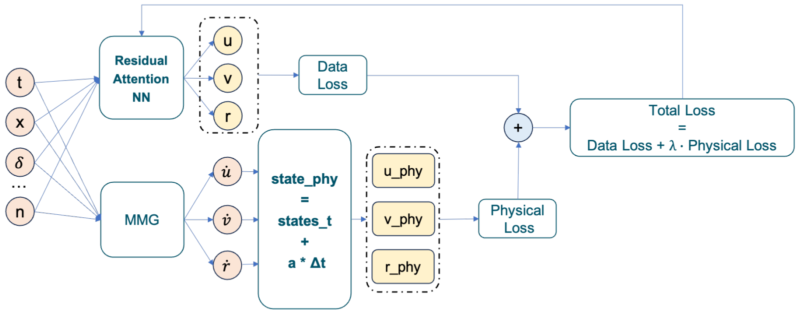


Figure 4: Physics-Informed Neural Network (PINN)

(1) Dual-path architecture. As illustrated in Fig. 4, the input features $(t, x, \delta, n, \dots)$ are simultaneously fed into:

- A neural network $\hat{\mathbf{y}} = [\hat{u}, \hat{v}, \hat{r}]$ to predict the next-step motion states;
- The MMG model to compute physical accelerations $(\dot{u}, \dot{v}, \dot{r})$ based on hydrodynamic equations.

These accelerations are numerically integrated to generate the physically predicted state:

$$[u_{\text{phy}}, v_{\text{phy}}, r_{\text{phy}}] = [u, v, r]_t + [\dot{u}, \dot{v}, \dot{r}] \cdot \Delta t$$

(2) Composite loss function. The network training objective includes both a data-driven regression loss and a physics-based consistency loss:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{data}} + \lambda \cdot \mathscr{L}_{\text{phy}} \tag{14}$$

- **Data loss** ($\mathscr{L}_{\text{data}}$): computed as Smooth L1 loss between predicted outputs and true states:

$$\mathscr{L}_{\text{data}} = \text{SmoothL1}([\hat{u}, \hat{v}, \hat{r}], [u, v, r])$$

- **Physics loss** ($\mathscr{L}_{\text{phy}}$): penalizes the deviation between predicted states and MMG-integrated physical predictions:

$$\mathscr{L}_{\text{phy}} = \text{SmoothL1}([\hat{u}, \hat{v}, \hat{r}],\ [u_{\text{phy}}, v_{\text{phy}}, r_{\text{phy}}])$$

(3) Adaptive weighting. To balance the learning from data and the adherence to physical principles, we introduce an adaptive weighting mechanism for the total loss function:

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{data}} + \lambda_t \cdot \mathscr{L}_{\text{phy}} \tag{15}$$

where λ_t dynamically adjusts the contribution of the physics-based loss $\mathscr{L}_{\text{phy}}$ relative to the data loss $\mathscr{L}_{\text{data}}$ during training. The weighting factor λ_t is updated at each training step based on the ratio of losses:

$$\lambda_{\text{target}} = \lambda_t \cdot (1 + \eta \cdot (\text{ratio} - 1)) \tag{16}$$

$$\lambda_{t+1} = \alpha \cdot \lambda_t + (1 - \alpha) \cdot \lambda_{\text{target}} \tag{17}$$

Here, ratio $= \mathscr{L}_{\text{data}}/\mathscr{L}_{\text{phy}}$, η is the adaptation rate (e.g., 0.002), and α is a smoothing factor (e.g., 0.98).

This formulation allows the model to adjust the strength of physical regularization automatically:

- If ratio > 1, indicating that the data loss dominates, then λ_t increases to strengthen the physical constraints.
- If ratio < 1, indicating that the physics loss dominates, then λ_t decreases to reduce the over-constraint.

This adaptive loss balancing strategy prevents either term from overwhelming the optimization, enhancing both convergence and generalization.

This grey-box formulation ensures that the neural network predictions not only match observed data but also remain consistent with the underlying ship dynamics. It improves robustness in noisy or unseen conditions and enables better generalization beyond the training domain.

2.2 Engine Power and Emission Modeling

In this section, our objective is to estimate the Specific Fuel Oil Consumption (SFOC) through a physically-informed, data-driven pipeline. Specifically, the motion states (u, v, r) and propeller speed n_p are first used to compute the effective engine power P_e based on propeller hydrodynamics and shaft transmission equations. Then, a neural network model is trained to map (P_e, ω_e) directly to SFOC, thus providing a continuous and differentiable approximation of the fuel efficiency landscape.

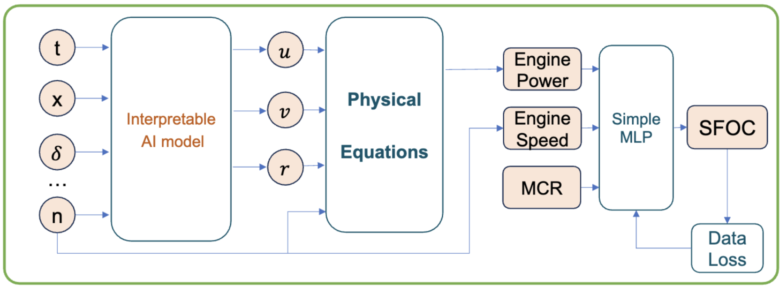


Figure 5: SFOC Estimate pipeline

2.2.1 Physical Derivation from Motion to Power

Specific Fuel Oil Consumption (SFOC) [g/kWh] of an engine is a parameter that reflects the efficiency of a combustion engine that burns fuel and produces rotational power, which is given by

$$\text{SFOC} = \frac{\dot{m}_{\text{fuel}}}{P_e} \tag{18}$$

where $\dot{m}_{\text{fuel}}$ is the instantaneous fuel mass flow rate (g/h) and P_e is the effective engine power (kW).

Although the definition of SFOC is rooted in fuel mass flow per unit power, the instantaneous fuel mass flow rate $\dot{m}_{\text{fuel}}$ is often difficult or impractical to measure accurately in real-time onboard systems.

Therefore, SFOC is typically modeled as a function of measurable engine variables, namely the engine power P_e (in kW) and the engine speed ω_e (in rpm). This functional relationship is often provided by the engine manufacturer in the form of certified contour maps. In our study, this empirical dependency is approximated using a data-driven neural network model:

$$\text{SFOC} = \mathscr{F}_{\text{NN}}(P_e, \omega_e, MCR)$$

Where MCR(Maximum Continuous Rating) is the maximum power point at which the ship's main engine can run stably for a long time under the rated design conditions. Different load points (that is, the ratio of actual power to MCR) correspond to different SFOC (specific fuel consumption rate).

The engine power P_e is calculated via:

$$P_e = \frac{P_p}{\eta_{\text{gear}} \cdot \eta_{\text{shaft}}} \tag{19}$$

where P_p is the propeller shaft power and $\eta_{\text{gear}}, \eta_{\text{shaft}}$ are the gear and shaft transmission efficiencies.

Propeller power is defined as:

$$P_p = \omega_p \cdot T_p \tag{20}$$

with $\omega_p = 2\pi n_p$ the angular velocity (rad/s), and T_p the propeller torque, computed from hydrodynamic relationships[21]:

$$T_p = K_Q(J_p) \cdot \rho \cdot n_p^2 \cdot D_p^5 \tag{21}$$

Here, $K_Q(J_p)$ is the torque coefficient, and J_p is the non-dimensional advance coefficient defined in equation 5. The relationship between K_Q and J_p is presented as Fif. 6 and can be approximated by a third-order polynomial regression based on empirical propeller charts[22]:

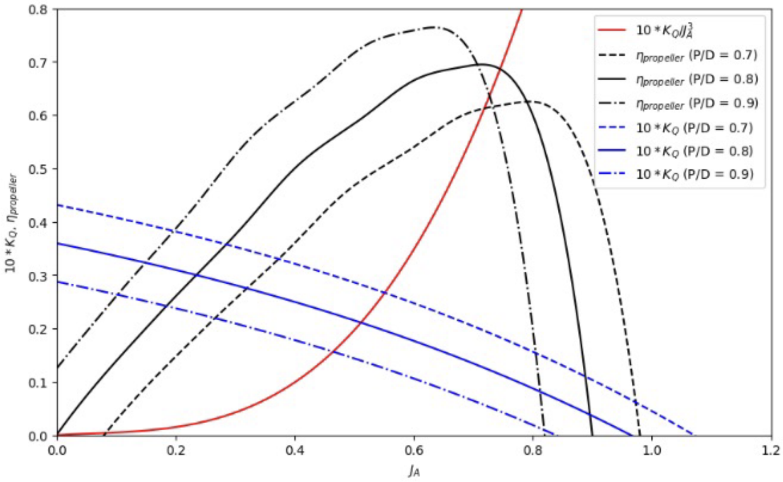


Figure 6: Use third-order polynomial regression to capture the specific relationship

$$K_Q(J_p) = k_3 J_p^3 + k_2 J_p^2 + k_1 J_p + k_0 \tag{22}$$

Substituting all the terms recursively, the effective engine power P_e can be expressed as a nonlinear function of motion state variables and control inputs:

$$P_e = f(u, n_p, D_p, \beta_p; \rho, K_Q(J_p), \eta)$$

Thus, the SFOC prediction task becomes a mapping from motion-derived physical quantities (e.g., P_e, n_p, MCR) to fuel consumption rate, which we subsequently model using a data-driven neural network in Section 2.2.3.

2.2.2 Dataset Rescaling via Froude Similarity

When operating on scaled-down model data (e.g., L3-model or L7-model of KVLCC2), all quantities are converted to full-scale values using Froude scaling laws and model-to-ship ratios to ensure physical consistency.

The SIMMAN2008 dataset is obtained from a 7-meter model of the KVLCC2 hull. To ensure physical consistency with real ship operations, all motion-related variables and power estimates must be mapped to full-scale values corresponding to a 320-meter vessel using Froude scaling laws.

Length Scale Ratio. The length scale ratio λ between the ship and the model is defined as:

$$\lambda = \frac{L_{\text{ship}}}{L_{\text{model}}} = \frac{320}{7} \approx 45.71 \tag{23}$$

Speed Conversion. According to Froude's law, linear velocity u scales with the square root of length:

$$u_{\text{ship}} = \lambda^{1/2} \cdot u_{\text{model}} \tag{24}$$

Given $u_{\text{model}} = 0.96 \sim 1.17$ m/s, the converted full-scale speed is:

$$u_{\text{ship}} \approx 6.5 \sim 7.9 \text{ m/s} \quad (\approx 12.6 \sim 15.3 \text{ knots}) \tag{25}$$

This aligns with the economic cruise speed of VLCCs, typically around $13 \sim 15.5$ knots.

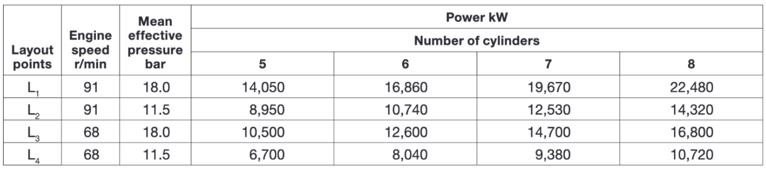
Rotational Speed Conversion. Propeller RPM scales with the inverse square root of length:

$$n_{\text{ship}} = \lambda^{-1/2} \cdot n_{\text{model}} \approx \frac{1}{\sqrt{45.71}} \cdot 614 \approx 90.8 \text{ RPM} \tag{26}$$

Power Conversion. Mechanical power scales with $\lambda^{3.5}$ based on geometric and dynamic similarity:

$$P_{\text{ship}} = \lambda^{3.5} \cdot P_{\text{model}} \approx 20 - 30 \text{ MW} \tag{27}$$

which falls within the realistic range of the VLCC propulsion requirements (20–30 MW), as seen in the power rating of S70MC6 engine family in Fig.7



Layout points	Engine speed r/min	Mean effective pressure bar	Power kW			
			Number of cylinders			
			5	6	7	8
L_1	91	18.0	14,050	16,860	19,670	22,480
L_2	91	11.5	8,950	10,740	12,530	14,320
L_3	68	18.0	10,500	12,600	14,700	16,800
L_4	68	11.5	6,700	8,040	9,380	10,720

Figure 7: The engine power falls in the 8 cylinders engine

SFOC Mapping. The full-scale engine operating points (P_e, ω_e) are then used to match the manufacturer-supplied SFOC map of the MAN-S70MC6 engine with a fixed-pitch propeller. Using the engine layout diagram, we align our power and speed values with operating points (e.g., L_1, L_2) and retrieve the SFOC via a data-driven regression.

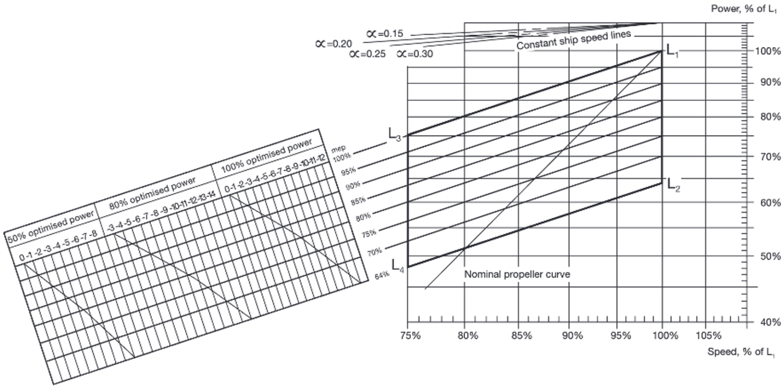


Figure 8: With different MCR, the engine correspond to different SFOC

This procedure ensures that the estimated SFOC corresponds to realistic, certified engine performance in full-scale VLCC operation.

2.2.3 Neural Network Estimation of SFOC

Given the physically computed engine power P_e and corresponding engine speed (RPM), a lightweight multi-layer perceptron (MLP) is trained to estimate the ship's Specific Fuel Oil Consumption (SFOC). The input vector to the network is defined as:

$$\mathbf{x}_{\mathrm{SFOC}} = [P_e,\ n_{\mathrm{RPM}},\ \mathrm{MCR}]$$

where P_e is the engine power, n_{RPM} is the shaft speed in revolutions per minute, and MCR is the maximum continuous rating used for normalization.

The network architecture consists of two to three fully connected layers with nonlinear activation (e.g., SiLU or ReLU), followed by a linear output layer that predicts SFOC in g/kWh. The training loss is defined as the mean squared error (MSE) between the predicted and measured SFOC:

$$\mathscr{L}_{\text{SFOC}} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i - y_i^{\text{true}}\right)^2 \tag{28}$$

This neural approximation provides a computationally efficient and differentiable module to estimate emission-related outputs from motion-physical inputs, enabling end-to-end carbon footprint estimation across the ship's operational trajectory.

3 Experiments and Results

3.1 Experimental Settings

3.1.1 Dataset and Input Features

The proposed models are evaluated using the SIMMAN2008 benchmark dataset, which contains free-running maneuvering trials of a 7m-scaled KVLCC2 model. The input to the neural network includes a sequence of physical motion states and control inputs:

$$\mathbf{x} = [t,\ x,\ y,\ \psi,\ u,\ v,\ r,\ p,\ \delta,\ nps]$$

Here, u, v, and r denote the surge velocity, sway velocity, and yaw rate, respectively; p denotes the roll velocity; ψ is the heading angle, δ is the rudder angle, and nps is the propeller shaft speed in revolutions per second (rps).

The target prediction variables are the next-step motion states $[\hat{u}, \hat{v}, \hat{r}]$, and in the extended pipeline, the estimated engine power P_e and the Specific Fuel Oil Consumption (SFOC).

3.1.2 Model Variants

To evaluate the impact of physical priors and residual learning, we compare the following model variants:

- **White-box (MMG)**: purely physical model with no learning component.
- **Black-box**: purely data-driven ResidualAttentionNN, trained directly on $[u, v, r]$ targets.
- **Black-box**: ResidualAttentionNN predicts residuals added to MMG outputs.
- **Grey-box**: Physical-informed Neural Network (PINN) trained with both data loss $\mathscr{L}_{\text{data}}$ and physics loss $\mathscr{L}_{\text{phy}}$.

All models are trained under the same optimizer (Adam), with Smooth L1 loss as the data criterion. The PINN variant uses an adaptive weighting scheme to balance $\mathscr{L}_{\text{data}}$ and $\mathscr{L}_{\text{phy}}$ during training.

3.1.3 Train/Test Split and Preprocessing

The SIMMAN2008 dataset is split into training and test sets, where entire trajectories from different maneuvering cases are used to prevent information leakage. The detailed split is as follows:

Table 3: Train/Test split of maneuvering cases in the SIMMAN2008 dataset

Split	**Maneuvering Case IDs**
Training Set	z1001, z1005, z1010P, z1010s, z1501, z2005, z2010s, z2505, z3005, z3505
Testing Set	z1505

All features are normalized using statistics (mean and standard deviation) computed from the training data only. This ensures fair evaluation and prevents test-time data leakage.

3.2 Motion State Prediction Results

3.2.1 Qualitative Visualization and Model Analysis

In this section, a side-by-side comparison of predicted surge velocity u, sway velocity v, and yaw rate r (in deg/s) across four different model types are presented: the MMG physical model (white-box), ResidualAttentionNN (pure data-driven black-box), ResidualCorrection (data-driven with physics baseline), and the proposed Physics-Informed Neural Network (PINN, grey-box). The visualizations are based on the same test set input and illustrate the temporal alignment between predicted state variables and actual measurements.

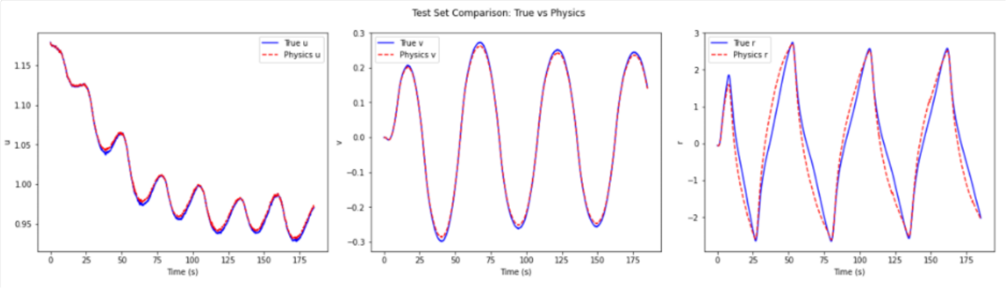


Figure 9: Prediction results of the MMG physical model for surge velocity (u), sway velocity (v), and yaw rate (r).

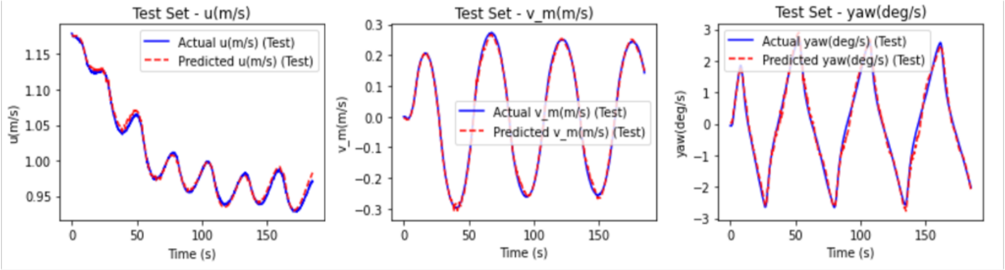


Figure 10: Prediction results of the pure data-driven model (ResidualAttentionNN) for surge velocity (u), sway velocity (v), and yaw rate (r).

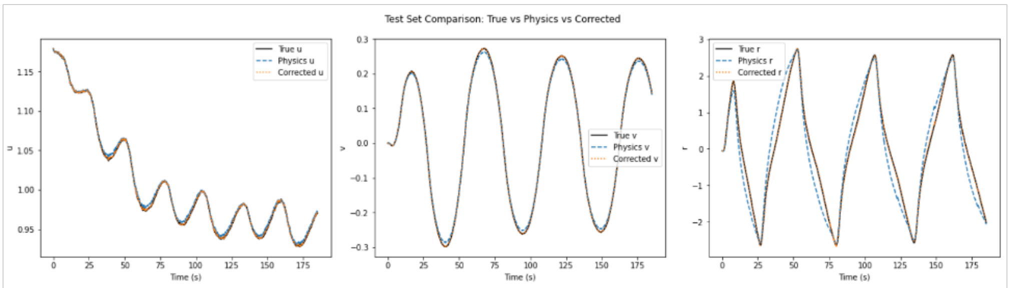


Figure 11: Prediction results of the ResidualAttentionNN of physical residuals correction for surge velocity (u), sway velocity (v), and yaw rate (r).

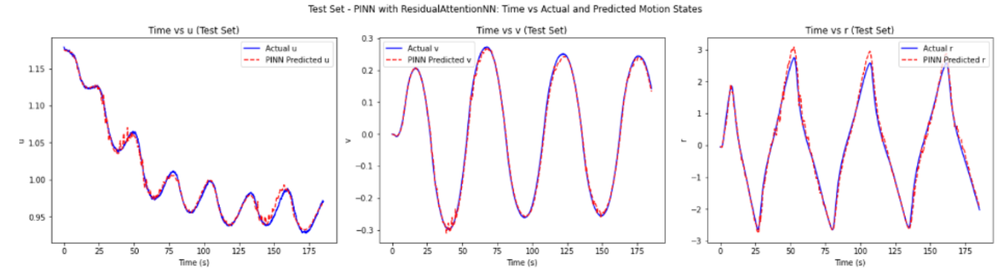


Figure 12: Prediction results of the proposed PINN model for surge velocity (u), sway velocity (v), and yaw rate (r).

From these results, all four models are able to reasonably capture the motion dynamics in terms of overall waveform and trend. In terms of prediction accuracy, the residual correction model exhibits the most consistent alignment with the ground truth across all three motion components, followed by the pure data-driven black-box model. The PINN performs slightly below the black-box model in accuracy but provides better physical consistency and stability. The MMG model, while physically interpretable, demonstrates clear deficiencies in yaw rate prediction due to unmodeled nonlinearities.

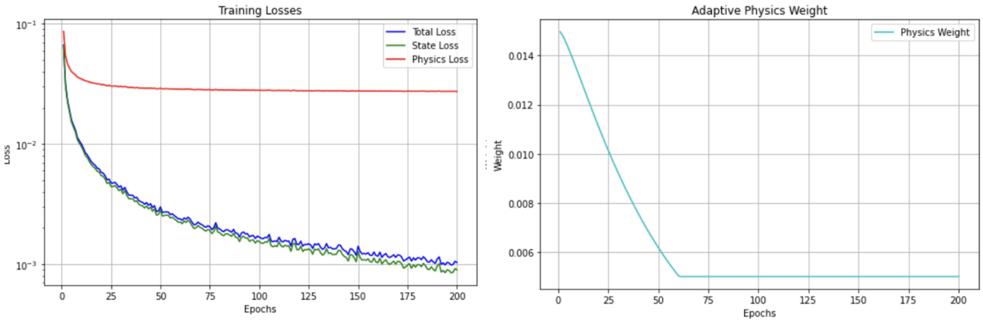


Figure 13: Training losses and adaptive physics weight during PINN optimization. Left: Total, state, and physics loss components. Right: Online adjustment of the physics loss weight λ_t.

Figure 13 illustrates the training dynamics of the PINN model over 200 epochs. The left subplot shows the evolution of three loss components: total loss $\mathscr{L}_{\text{total}}$, state loss $\mathscr{L}_{\text{data}}$, and physics loss $\mathscr{L}_{\text{phy}}$. It can be observed that both the total and data losses decrease steadily as training progresses, indicating successful data fitting. The physics loss, in contrast, remains relatively flat, suggesting that the model maintains a consistent level of physical regularization throughout training.

The right subplot shows the evolution of the adaptive physics weight λ_t, which is adjusted online using a dynamic weighting strategy. Initially, the weight is higher to encourage physical consistency. As the data loss $\mathscr{L}_{\text{data}}$ decreases and becomes dominant, λ_t is gradually reduced, stabilizing at a lower value after approximately 60 epochs. This adaptive mechanism allows the network to focus more on data fitting in the later stages of training while preserving physical plausibility.

In conclusion, the ResidualAttentionNN with physical residual correction achieves the highest numerical accuracy among all models, while still maintaining a certain degree of physical plausibility due to its reliance on the MMG baseline. The purely data-driven ResidualAttentionNN also delivers strong predictive performance, but lacks interpretability and offers no guarantees of physical consistency. The MMG model, while physically interpretable and reasonably accurate in u and v, performs less effectively in predicting yaw rate due to unmodeled nonlinearities. The proposed PINN model offers a balanced trade-off between accuracy and interpretability by incorporating physics-informed regularization, leading to stable and trustworthy predictions. Therefore, both the Residual Correction model and the PINN are favorable choices when high prediction accuracy and physical insight are simultaneously required.

3.2.2 Robustness Comparison under Noisy Conditions

To further evaluate the robustness of different models under sensor uncertainty, we introduce synthetic Gaussian noise into the input motion data with varying noise levels ranging from 2% to 5%. In this section, we compare the performance of the Residual Correction model and the Physics-Informed Neural Network (PINN) when subject to 5% input noise.

Figure 14 and Figure **??** show the predictive performance of both models on the test motion state `z1505` under 5% input noise. The Residual Correction model exhibits substantial degradation in its output curves, where high-frequency noise is clearly propagated through the learned residual pathway. In contrast, the PINN model demonstrates superior robustness, effectively suppressing noise propagation and maintaining smooth, accurate motion estimates for all three states (u, v, and r).

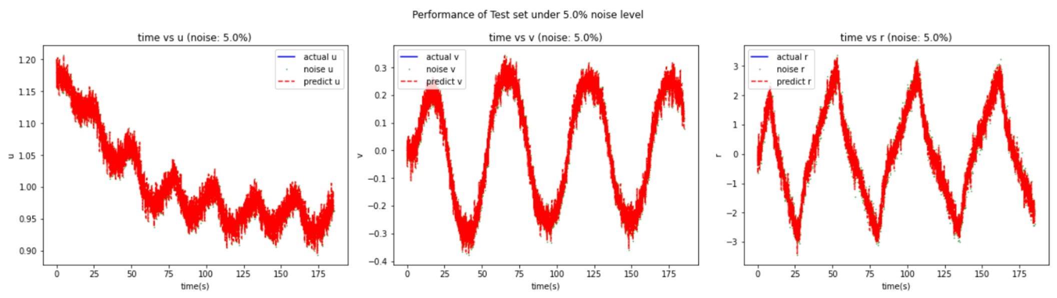


Figure 14: Robustness of Residual Correction model under 5% input noise. Predictive curves are visibly affected by high-frequency disturbances.

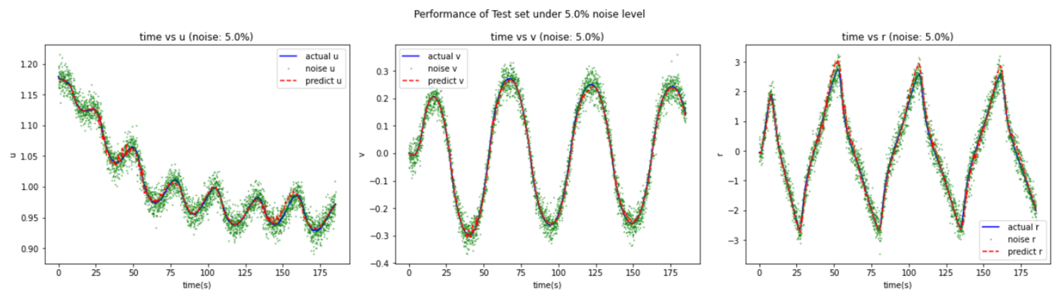


Figure 15: Robustness of PINN model under 5% input noise. Predictions remain smooth and close to ground truth across all motion components.

To quantitatively compare the robustness performance, Table 4 summarizes four error metrics (MSE, RMSE, MAE, and MAPE) on the u component across different noise levels. The PINN consistently outperforms the Residual Correction model at all noise intensities, demonstrating its superior generalization and resistance to overfitting in the presence of perturbations.

Table 4: Quantitative robustness evaluation for $\boldsymbol{u}$ prediction under various noise levels. Lower values indicate better performance.

Noise Level	Model	MSE	RMSE	MAE	MAPE (%)
2%	Residual	2.49e-05	0.00499	0.00398	0.3976
	PINN	**2.11e-05**	**0.00459**	**0.00332**	**0.3335**
3%	Residual	5.65e-05	0.00752	0.00599	0.5982
	PINN	**2.11e-05**	**0.00459**	**0.00332**	**0.3335**
4%	Residual	0.00010	0.01	0.00798	0.7978
	PINN	**2.11e-05**	**0.00459**	**0.00332**	**0.3335**
5%	Residual	0.00016	0.0126	0.01	1.0
	PINN	**2.11e-05**	**0.00459**	**0.00332**	**0.3335**

In summary, the PINN model demonstrates stronger robustness than the Residual Correction model, both qualitatively and quantitatively. This validates the contribution of physics-based regularization in enhancing model generalization and denoising capacity, particularly under operational noise environments.

3.3 SFOC Prediction Results

The dataset used for SFOC prediction is derived from the manufacturer specifications of the MAN-S70MC6 engine, incorporating a variety of operating conditions. A heatmap of pairwise correlations among variables is shown in Figure 16, highlighting moderate positive correlation between engine power and SFOC, and a middle correlation between MCR and engine speed. Based on this analysis, three selected input features—engine power ($\boldsymbol{P_e}$), engine speed (ω_e), and MCR—are fairly reasonable.

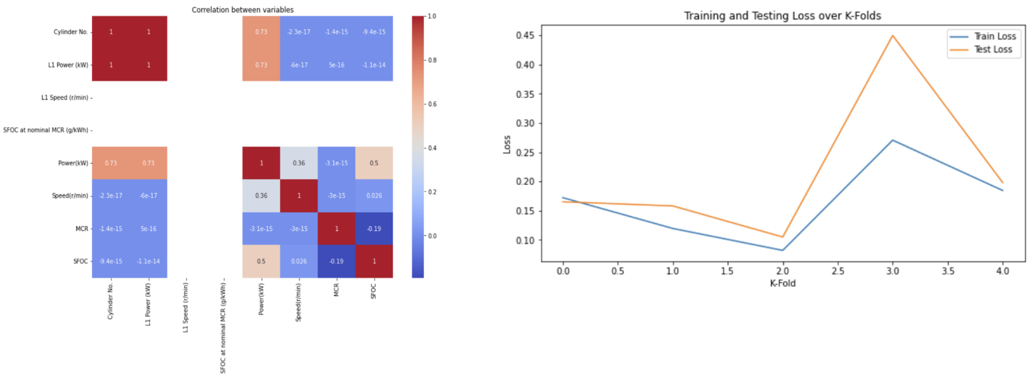


Figure 16: Left: Feature correlation heatmap; Right: K-fold training and testing loss.

To assess the generalization ability of the model, a 5-fold cross-validation experiment is performed. The testing loss across folds remains low and consistent, with a prediction error ranging from 0.18% to 0.39%. This indicates that the model is both accurate and robust, capable of learning the mapping from propulsion-induced variables to SFOC effectively, even under varying data partitions.

4 Conclusion

This study presents a comprehensive hybrid modeling framework that integrates physical modeling and data-driven learning to achieve accurate, interpretable, and robust predictions of ship dynamics and carbon emissions. Targeting the critical need for low-carbon maritime solutions, our approach enables full-process estimation—from vessel motion to propulsion modeling and ultimately to fuel efficiency assessment.

Starting from the physical modeling side, we adopt the MMG (Maneuvering Modeling Group) standard to simulate ship motion, explicitly capturing hull hydrodynamic forces, rudder lift, and propeller thrust. A complete propulsion-energy chain is constructed, linking the propeller model, the diesel engine model, and the fuel consumption model. This results in a unified physical structure that enables estimation of engine power and fuel consumption directly from navigational states.

To address modeling inaccuracies in ship motion prediction, this study introduces Black-box models ResidualAttentionNNs: one for pure data-driven, and the other one learns residual errors between physics-based MMG predictions and observed data. Building on this, a grey-box Physics-Informed Neural Network (PINN) is developed by incorporating physical consistency constraints derived from MMG dynamics, with adaptive weighting to balance physical laws and data-driven objectives. The first major finding is that this hybrid model significantly enhances robustnes under input noise, while maintaining high accuracy and smoothness.

The second key finding lies in the accurate estimation of Specific Fuel Oil Consumption (SFOC). A simple MLP network, trained on propulsion-derived inputs—engine power, engine speed, and MCR—achieves outstanding performance, with prediction errors consistently controlled within 0.18%–0.39%. This confirms the model's capacity to capture complex nonlinear fuel behavior with minimal structural complexity, enabling efficient end-to-end integration into a carbon estimation pipeline.

In summary, this work establishes a full-process, physics-guided learning pipeline for maritime motion and emission prediction. It effectively links control inputs to final carbon emissions via interpretable intermediate layers, offering a scalable and robust foundation for green and autonomous ship operations.

5 Future Work

This study has proposed a physics-informed hybrid modeling framework for ship motion prediction and fuel consumption estimation, laying a solid foundation for interpretable and robust carbon-aware maritime control. Nonetheless, several limitations remain to be addressed, and future work will focus on enhancing the model's practicality, adaptability, and scalability.

While the current framework is built on simulation-based datasets such as SIMMAN2008, the next stage of development will incorporate real-world data from Automatic Identification System (AIS) logs, onboard sensors, and digital twin platforms. These time-series datasets will allow the framework to be validated and fine-tuned under realistic operational conditions, enabling more reliable carbon emission estimation in practice.

To further enhance robustness, we aim to incorporate environmental disturbances into the modeling process. External factors such as waves, wind, currents, and shallow water effects can significantly impact ship dynamics. Embedding these components into the physics-informed structure is expected to improve stability and prediction quality under real-world sea conditions.

In terms of generalization, extending the framework beyond the KVLCC2 tanker to other vessel types—such as container ships, bulk carriers, and LNG-fueled ships—will be a crucial step. Such extension requires adaptation of both the physical modeling components and the neural architecture to reflect different hull geometries, propulsion systems, and fuel characteristics.

We also plan to strengthen the integration of physical constraints. Additional regularization strategies—such as mass and momentum conservation, first- and second-order derivative penalties, and temporal smoothing—will be explored to enforce consistency with known dynamics. These constraints are expected to improve interpretability and reduce overfitting, particularly when the model is deployed across different operating profiles.

To further expand the application scope of the proposed modeling framework, future work will focus on integrating it with advanced control algorithms. By coupling the motion and emission prediction modules with model predictive control (MPC) or reinforcement learning-based eco-control strategies, the system can provide real-time, trajectory-aware decision-making for energy optimization. Such integration will allow the model not only to simulate and predict ship behavior, but also to actively guide control inputs that minimize fuel consumption and emissions under dynamic environmental and operational constraints.

Altogether, these developments will help transition the proposed framework from a simulation-validated prototype to a deployable solution for carbon-efficient and intelligent maritime operations.

References

[1] S IMO. *Fourth greenhouse gas study 2020*. 2020.

[2] International Maritime Organization. *EEXI—Energy Efficiency Existing Ship Index*. MEPC.335(76). 2021.

[3] International Maritime Organization. *CII—Carbon Intensity Indicator*. MEPC.336(76). 2021.

[4] Juan Moreno-Gutiérrez et al. "Comparative analysis between different methods for calculating on-board ship's emissions and energy consumption based on operational data". In: *Science of The Total Environment* 650 (2019), pp. 575–584. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2018.09.045. URL: https://www.sciencedirect.com/science/article/pii/S0048969718334636.

[5] Ícaro Aragão Fonseca et al. "A Standards-Based Digital Twin of an Experiment with a Scale Model Ship". In: *Computer-Aided Design* 145 (2022), p. 103191. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2021.103191. URL: https://www.sciencedirect.com/science/article/pii/S001044852100186X.

[6] Thor I Fossen. "Motion control systems". In: *Handbook of Marine Craft Hydrodynamics and Motion Control* (2011), pp. 343–415.

[7] Thor I. Fossen and Tor A. Johansen. "A Survey of Control Allocation Methods for Ships and Underwater Vehicles". In: *2006 14th Mediterranean Conference on Control and Automation*. 2006, pp. 1–6. DOI: 10.1109/MED.2006.328749.

[8] Hironori Yasukawa and Yasuo Yoshimura. "Introduction of MMG standard method for ship maneuvering predictions". In: *Journal of Marine Science and Technology* 20 (Mar. 2014), pp. 37–52. DOI: 10.1007/s00773-014-0293-y.

[9] Volker Bertram. *Practical ship hydrodynamics*. Elsevier, 2012.

[10] Kun Dai and Yunbo Li. "Manoeuvring prediction of KVLCC2 with hydrodynamic derivatives generated by a virtual captive model test". In: *Polish Maritime Research* (2019).

[11] Zihao Wang et al. "Kernel-based support vector regression for nonparametric modeling of ship maneuvering motion". In: *Ocean Engineering* 216 (2020), p. 107994. ISSN: 0029-8018. DOI: https://doi.org/10.1016/j.oceaneng.2020.107994. URL: https://www.sciencedirect.com/science/article/pii/S0029801820309434.

[12] Gollavelli Rajesh and Subrata K Bhattacharyya. "System identification for nonlinear maneuvering of large tankers using artificial neural network". In: *Applied Ocean Research* 30.4 (2008), pp. 256–263.

[13] Hong-Wei He et al. “Nonparametric modeling of ship maneuvering motion based on self-designed fully connected neural network”. In: *Ocean Engineering* 251 (2022), p. 111113.

[14] Mingyang Zhang et al. “A deep learning method for the prediction of ship fuel consumption in real operational conditions”. In: *Engineering Applications of Artificial Intelligence* 130 (2024), p. 107425.

[15] Xin-Guang Zhang and Zao-Jian Zou. “Black-box modeling of ship manoeuvring motion based on feed-forward neural network with Chebyshev orthogonal basis function”. In: *Journal of marine science and technology* 18 (2013), pp. 42–49.

[16] Hong-Wei He and Zao-Jian Zou. “Black-box modeling of ship maneuvering motion using system identification method based on BP neural network”. In: *International Conference on Offshore Mechanics and Arctic Engineering*. Vol. 84386. American Society of Mechanical Engineers. 2020, V06BT06A037.

[17] Kamaljyoti Nath et al. “Physics-informed neural networks for predicting gas flow dynamics and unknown parameters in diesel engines”. In: *Scientific Reports* 13.1 (2023), p. 13683.

[18] George Em Karniadakis et al. “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6 (2021), pp. 422–440.

[19] Tongtong Wang, Robert Skulstad, and Houxiang Zhang. “Physics-Informed Neural Networks for Robust System Identification of Ship Roll Dynamics With Noise Resilience”. In: *IEEE Transactions on Industrial Informatics* (2025).

[20] Shengze Cai et al. “Physics-informed neural networks (PINNs) for fluid mechanics: A review”. In: *Acta Mechanica Sinica* 37.12 (2021), pp. 1727–1738.

[21] Huilin Ren, Yu Ding, and Congbiao Sui. “Influence of EEDI (Energy Efficiency Design Index) on ship–engine–propeller matching”. In: *Journal of Marine Science and Engineering* 7.12 (2019), p. 425.

[22] Huilin Ren, Yu Ding, and Congbiao Sui. “Influence of EEDI (Energy Efficiency Design Index) on ship–engine–propeller matching”. In: *Journal of Marine Science and Engineering* 7.12 (2019), p. 425.