# Physics-Informed Neural Networks for Vessel Trajectory Prediction: Learning Time-Discretized Kinematic Dynamics via Finite Differences

Md Mahbub Alam
Faculty of Computer
Science
Dalhousie University
Halifax, Nova Scotia
Canada
mahbub.alam@dal.ca

Amilcar Soares
Linnaeus University
Växjö, Sweden
amilcar.soares@lnu.se

José Fernando
Rodrigues-Jr
University of Sao Paulo
Sao Carlos, Brazil
junio@icmc.usp.br

Gabriel Spadon*
Faculty of Computer
Science
Dalhousie University
Halifax, Nova Scotia
Canada
gabriel@spadon.com.br

## Abstract

Accurate vessel trajectory prediction is crucial for navigational safety, route optimization, traffic management, search and rescue operations, and autonomous navigation. Traditional data-driven models lack real-world physical constraints, leading to forecasts that violate vessel motion dynamics, such as in scenarios with limited or noisy data where sudden course changes or speed variations occur due to external factors. To address this limitation, we propose a Physics-Informed Neural Network (PINN) approach for trajectory prediction that integrates a streamlined kinematic model for vessel motion into the neural network training process via first- and second-order, finite-difference physics-based loss functions. These loss functions, discretized using the first-order forward Euler method, Heun's second-order approximation, and refined with a midpoint approximation based on Taylor series expansion, enforce fidelity to fundamental physical principles by penalizing deviations from expected kinematic behavior. We evaluated PINN using real-world AIS datasets that cover diverse maritime conditions and compared it with state-of-the-art models. Our results demonstrate that the proposed method reduces average displacement errors by up to 32% across models and datasets while maintaining physical consistency. These results enhance model reliability and adherence to mission-critical maritime activities, where precision translates into better situational awareness in the oceans.

## CCS Concepts

• **Information systems** → **Location based services**; • **Computing methodologies** → **Neural networks**; • **Applied computing** → **Transportation**.

## Keywords

AIS Data, Physics-Informed Neural Networks, Trajectory Prediction, Maritime Navigation

---

*Corresponding Author

## 1 INTRODUCTION

Enhancing situational awareness on ocean transportation is key for preventing collisions, a risk that has grown significantly due to the substantial increase in maritime traffic in recent years [1, 2, 27]. Beyond improving situational awareness, the accurate prediction of future vessel movements is critical for a wide range of maritime applications. These include enabling efficient navigation through route optimization, reducing fuel consumption and transit times [30]; supporting proactive traffic management to alleviate congestion and improve safety in busy waterways [29, 36]; informing timely and effective responses during maritime emergencies, such as search and rescue operations; aiding in the detection and tracking of vessels involved in illicit activities, thereby contributing to maritime security [18]; and providing vital input for the safe and reliable operation of autonomous vessels [5]. Trajectory prediction has emerged as an extensively explored research area in the maritime domain, driving innovation across applications [11, 31].

Innovative approaches are being developed to address the challenges of accurate trajectory prediction, particularly using machine learning techniques. One significant recent advancement in machine learning is the Physics-Informed Neural Networks (PINNs), well-suited for applications governed by physical principles [5, 26], such as vessel trajectory prediction. Unlike purely data-driven models that rely solely on observed data, PINNs embed domain-specific physical laws into the neural network training process, improving data efficiency, enhancing generalization, and ensuring physically consistent predictions. As vessel movement follows kinematic principles, PINNs can integrate these dynamics into the model, leading to accurate and reliable forecasts.

Over the past decade, maritime trajectory prediction research has shifted from relying solely on physical principles to adopting data-driven models, leveraging classical machine learning and, more recently, deep learning [11, 29, 31]. This transition has been particularly impactful in recent years, with deep learning models achieving significant improvements in prediction accuracy. These

advancements began with RNN-based architectures such as LSTMs and GRUs [4, 20, 24], progressed to CNNs [19], and now encompass more advanced models such as Temporal Convolutional Networks (TCNs) [13], Graph Neural Networks (GNNs) [30], and Transformers [12, 15]. The initial objective was to improve prediction accuracy by using large datasets and introducing novel approaches through architectural changes or model fusion [9, 28]. Additionally, some approaches incorporate a distance-based loss function to minimize the displacement between predicted and actual trajectories [7, 32]. However, in scenarios involving limited, sparse, or noisy data, the accuracy of these models remains limited, especially when sudden course changes or speed variations arise due to environmental factors. PINNs offer a promising solution by integrating physical laws with observed data, enabling them to achieve more accurate and reliable predictions, even under challenging conditions.

Moreover, since the International Maritime Organization (IMO) mandated the adoption of the Automatic Identification System (AIS) in 2004 [21] and its subsequent satellite integration in 2008 [16], AIS data has become integral to maritime research. Particularly, dynamic AIS messages containing crucial kinematic information such as latitude, longitude, speed over ground (SOG), and course over ground (COG) have been extensively leveraged for diverse applications, including vessel trajectory prediction. While predominantly data-driven approaches have utilized these core AIS features as inputs for prediction models [31], some methods have further incorporated derived kinematic attributes — including acceleration, jerk, COG rate, and bearing — to capture complex vessel mobility patterns [1]. However, a critical gap remains: the lack of integrated modeling of kinematic constraints inherently embedded in AIS data, which fundamentally governs real-world vessel dynamics. Our work addresses this gap by integrating vessel kinematic constraints into data-driven models through a PINN framework.

This paper proposes a PINN framework that leverages first- and second-order finite-difference physics-based loss functions to enhance vessel trajectory prediction using AIS data. These loss functions, derived from a simplified kinematic model of vessel motion, enforce physical consistency by penalizing deviations from physically expected kinematic behavior. The model is discretized using a first-order forward Euler method and Heun's second-order approximation, implemented with forward finite differences. To further improve accuracy, a midpoint approximation based on a first-order Taylor series expansion is also incorporated into the calculation of expected displacement. Importantly, the framework is model-agnostic, allowing seamless integration with state-of-the-art deep learning architectures, including RNNs, CNNs, ConvLSTM, TCNs, and Transformers, for vessel trajectory prediction. The main contributions are summarized as follows:

- We propose a novel PINN approach incorporating physics constraints into neural network training via Euler's first-order and Heun's second-order forward finite-difference loss functions, using a mid-point approximation based on a Taylor series expansion to refine the expected displacement.
- We benchmark the PINN framework on widely used sequence models — LSTMs, GRUs, CNNs, ConvLSTMs, TCNs, and Transformers — exhibiting its model-agnostic nature.

- We assessed PINN efficacy by varying the model complexity, prediction horizons, and approximation order, while factoring in navigational area complexity.
- We validate prediction accuracy by selecting models that balance physical plausibility with optimal convergence, rather than focusing solely on minimizing displacement error.

The remainder of this paper is organized as follows: Section 2 reviews recent related work; Section 3 outlines the methodology and proposal formulation; Section 4 presents and analyzes the experimental results; and Section 5 concludes with final remarks.

## 2 RELATED WORK

With the surge of studies in physics-driven machine learning, several review studies explored its applicability across domains [5, 26]. However, in the intersection of ocean mobility, maritime system reliability, and shipping operations, the meeting between PINNs and their applications remains underexplored. This gap exists largely due to the lack of direction and synthesis needed to support future research, define sound baselines, and map the landscape of physics-guided learning in ship movements and operations. To build such a foundation, we categorize the literature into groups based on how physical knowledge is integrated into models:

(1) **Physics as Structure:** models that directly embed physical laws (*i.e.*, equations) into their architectures, ensuring that vessel trajectories conform to established physical principles;

(2) **Physics as Direction:** hybrid methods that enrich data-driven models with features, priors, or multi-modal signals, blending domain knowledge with learning from data; and,

(3) **Physics as Correction:** approaches that use physics to regularize or correct predictions, such as guiding generative processes or refining numerical outputs to achieve realistic and reliable predictions.

Within these categories, detailed subsequently, current models often incorporate only partial physics. Many physics-informed approaches utilize environmental data or simple motion constraints but have not yet fully captured the complexities of ship dynamics (*e.g.*, hull hydrodynamics and six-degree-of-freedom motions[1]). Fully integrating high-fidelity naval architecture models into neural networks remains challenging due to computational complexity and the requirement for specialized expert knowledge. Furthermore, working with AIS data inherently poses constraints, as it typically lacks detailed kinematic and dynamic measurements.

This paper serves as a primer on applying PINNs to AIS data for ocean mobility and related applications. Current AIS-based research predominantly employs 2-DoF models, ignoring the z-axis, although scenarios like shallow waters or vessel drafts may require 3-DoF. The integration of full 6-DoF motion remains limited. While research continues to evolve, this work takes a significant step by introducing the first theoretical framework that integrates kinematic physics into neural networks for trajectory modeling, with the aim of enhancing model applicability, usability, and accuracy.

---

[1]**6-DoF:** *Surge* (forward/backward), *Sway* (left/right), *Heave* (up/down), *Roll* (rotation around X-axis), *Pitch* (rotation around Y-axis), and *Yaw* (rotation around Z-axis).

(1) *Structure* – **Models Constrained by Equations**

Zhao *et al.* (2024) [35] propose a framework for Autonomous Underwater Vehicles (AUV) that integrates PINNs with classical dynamic equations to capture the full 6-DoF motion of AUVs. The study embeds the spatial maneuvering motion equations directly into the loss function of a fully connected neural network, which is trained via a multi-step iterative process using a fourth-order Runge–Kutta scheme for time integration. Simulation experiments demonstrate that PINN yields stable, accurate, generalized long-term motion prediction even with limited training data. Field tests on a micro-AUV confirm effective trajectory tracking with control errors below one degree.

Papandreou *et al.* (2025) [17] propose an interpretable model that augments traditional physics-based motion prediction by optimizing hydrodynamic parameters via constrained nonlinear least squares. The model integrates a 3-DoF[2] physics-based framework (incorporating rudder and propeller forces and vessel resistance) with data-driven parameter tuning to capture ship-specific behaviors. Eleven parameters, including those governing the resistance polynomial and rudder force coefficients, are estimated using synthetic trajectory data derived from realistic ship maneuvers. Validation on datasets of two container ships demonstrates that fitted models predict trajectories with $51.6 - 57.8\%$ higher accuracy and $72.36 - 89.67\%$ greater consistency than conventional baselines.

Mathioudakis *et al.* (2025) [14] present a three-dimensional physics-based model for ship motion prediction of long container vessels. The model integrates dynamic equations for surge, sway, and yaw with hydrodynamic derivative methods to compute forces from control inputs (rudder and propeller) and environmental effects (wind, waves, and currents) via numerical integration. Validation against a baseline model and sea trial data shows the model can replicate vessel trajectories under typical conditions while revealing challenges during transient maneuvers with small rudder angles.

(2) *Direction* – **Hybrid and Augmented Learning**

Lang, Wu, and Mao (2024) [10] present a physics-informed grey-box model for ship speed prediction, combining a physics-based (PINN) and a data-driven (XGBoost) component. The methodology uses parallel modeling, where PINNs estimate the expected calm water speed from propulsion power and draft using speed-power model tests, whereas XGBoost predicts speed reduction under current ocean conditions. Validation with full-scale data from a chemical tanker demonstrates that the model achieves a 30% better prediction accuracy than a traditional black-box model and reduces the arrival time error by about 50%.

Chen *et al.* (2025) [3] propose a hybrid approach that combines LSTM with a PSO-GWO (*i.e.*, swarm-based meta-heuristic) algorithm for short-time ship trajectory prediction. The model captures the nonlinear and time-varying characteristics of vessel motion through long-term dependencies in AIS data. Results indicate that this approach achieves lower mean absolute and squared errors while reducing the optimization time and enhancing predictive accuracy and efficiency.

Guo et al. (2025) [8] introduce a vessel-influenced LSTM model for trajectory prediction, incorporating vessel influence maps to capture the effects of surrounding vessels. The model integrates vessel motion, environmental, and static factors with the influence of neighboring vessels while also using Gaussian prediction combined with Monte Carlo dropout to estimate uncertainty. A temporally weighted hybrid loss function is proposed to balance prediction accuracy with uncertainty quantification. Experiments on AIS data from Galveston Bay, USA, show that the model achieves lower mean distance errors than baselines on standard and unseen test sets, particularly under complex maritime conditions.

Suo, Ding, and Zhang (2024) [25] propose a deep-learning framework based on Mamba for ship trajectory prediction that employs a selective state-space model to process long sequential data efficiently. The model integrates hardware-aware state expansion and a simplified architecture to overcome the limitations of conventional methods such as LSTM, GRU, and Transformer. The model improved prediction accuracy, inference speed, and resource utilization on AIS data from the Beijing–Hangzhou Canal.

Zhao *et al.* (2025) [34] propose a deep learning framework that integrates inter-ship interactions and navigational uncertainties into the prediction model. It is built upon an encoder-decoder LSTM architecture that incorporates three key attention-based modules: a Position Attention Block that captures mutual positional influences among vessels, an Information Fusion Block that integrates differential navigation state information to represent uncertainty, and a Global Attention Block that dynamically aligns encoder outputs with the decoder's context. The experimental results on AIS datasets from multiple maritime regions demonstrate that the model outperforms baseline models.

Song et al. (2024) [22] propose the Transformer Gravity model, a gravity-inspired deep learning framework to forecast global maritime traffic flows and enhance risk assessments for non-indigenous species spread via ballast water. The model extends traditional gravity formulations by integrating features, such as shipping flux density, geodesic distances, bilateral trade volumes, and graph-based centrality metrics, into a Transformer architecture that effectively captures short- and long-term dependencies in vessel movement. Evaluated on global shipping networks derived from AIS data (2017–2019), this approach demonstrates over 10% higher prediction accuracy compared to conventional deep-gravity and regression models.

(3) *Correction* – **Generative and Precision-Critical Models**

Zhang *et al.* (2025) [33] propose a diffusion probabilistic framework for long-term vessel trajectory imputation that tackles extensive AIS data gaps. Their model uses a pre-trained trajectory embedding block to extract movement patterns and a transformer encoder to condition the reverse denoising process, generating continuous, multi-point imputed trajectories. A physics-guided discriminator enforces kinematic constraints between positional and angular data and curbs cumulative error. Experiments on a real-world AIS dataset demonstrate that this approach recovers long-term vessel trajectories with improved accuracy compared to existing imputation methods.

Ferreira and Campbell (2025) [6] introduce an RNN architecture that incorporates a Decimal Preservation (DP) layer to capture

---

[2] **3-DoF:** *Surge* (forward/backward), *Sway* (left/right), and *Heave* (up/down).

minute latitude and longitude variations and mitigate floating-point rounding errors. Against the Ornstein–Uhlenbeck baseline, the proposed architecture reduces the prediction errors of cargo vessels by up to 50% and demonstrates that the DP layer enhances the performance of Elman's RNN, LSTM, and GRU models.

## 3 METHODOLOGY

This section defines the trajectory prediction problem and details the proposed PINN architecture.

### 3.1 Preliminaries

*Trajectory*. A trajectory $\mathcal{T}$ is a sequence of points generated by a moving object. Each point $p_t$ in the trajectory represents the object's coordinates $(x_t, y_t)$ and its kinematic state at time $t$:

$$\mathcal{T} = \{p_1, p_2, \ldots, p_n\} \tag{1}$$

where $n$ is the number of temporal observations in the trajectory. For vessel trajectories, each point $p_t = (x_t, y_t, v_t, \psi_t, a_t, \dot{\psi}_t)$ is characterized by the following features at time $t$:

- $x_t$: Latitude (in degrees)
- $y_t$: Longitude (in degrees)
- $v_t$: Speed Over Ground (SOG, in meters per second)
- $\psi_t$: Course Over Ground (COG, in degrees)
- $a_t$: Acceleration (in meters per second squared)
- $\dot{\psi}_t$: rate of change of COG (in degrees per second)

Accordingly, for a point $p_t$, the input feature vector for the neural network model is defined as its transpose $p_t{}^T = \mathbf{x}_t$:

$$\mathbf{x}_t = [x_t, y_t, v_t, \psi_t, a_t, \dot{\psi}_t]^T \tag{2}$$

*Trajectory Prediction*. Trajectory prediction is the task of forecasting the future trajectory with respect to a point of interest $p_t$. That is, $\mathcal{T}_{\text{pred}} = \{\hat{p}_{t+1}, \hat{p}_{t+2}, \ldots, \hat{p}_{t+H}\}$, of a moving object, given its observed trajectory $\mathcal{T}_{\text{obs}} = \{p_1, p_2, \ldots, p_t\}$. $H$ is the prediction horizon, indicating the number of future time steps to be predicted.

### 3.2 PINN Architecture

The principle of PINNs is to train a neural network to learn from observed data and satisfy underlying physical equations. A PINN, denoted $f_\theta(\mathbf{x}_t)$, acts as a function approximator, learns a mapping from the input feature vector $\mathbf{x}_t = p_t{}^T$ to the predicted output $\hat{\mathbf{y}}_{t+i} = [\hat{x}_{t+i}, \hat{y}_{t+i}]^T$, which is a vector of latitude and longitude at future time $t + i$, for $i = 1, 2, \ldots, H$.

To train the PINN, we minimize a loss function, as depicted in Figure 1. This function is a weighted sum of two distinct terms: the data loss and the physics loss. The *data loss* ensures the model's predictions align with observed trajectory data, while the *physics loss* enforces adherence to the underlying kinematic constraints governing vessel motion. This dual objective allows PINNs to complement data-driven approaches by incorporating physical consistency into their predictions.

*Data Loss*. The data loss component, $\mathcal{L}_{\text{data}}(\theta)$, quantifies the discrepancy between the predicted trajectory and the ground truth data; to that end, we use the Mean Squared Error (MSE):
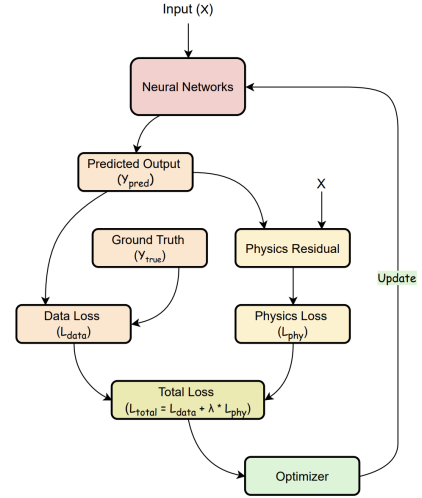


**Figure 1: The PINN's training framework optimizes the model by integrating data and physics losses to guide it towards physically informed trajectory predictions. PINN training is an iterative optimization process. In each iteration, the network generates predictions based on input data. A total loss is computed by comparing these predictions to ground truth and physical laws, respectively. The optimizer then minimizes the loss by updating the network's parameters, repeating until convergence.**

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{H} \left\| \mathbf{y}_{i,t+j} - \hat{\mathbf{y}}_{i,t+j} \right\|^2 \tag{3}$$

where:

- $N$ is the batch size, representing the number of trajectories used in each training iteration.
- $H$ is the prediction horizon.
- $\mathbf{y}_{j,t+i} = [x_{j,t+i}, y_{j,t+i}]^T$ is the ground truth latitude and longitude vector for the $j$-th trajectory at time $t + i$.
- $\hat{\mathbf{y}}_{j,t+i} = f_\theta(\mathbf{x}_{j,t}) = [\hat{x}_{j,t+i}, \hat{y}_{j,t+i}]^T$ is the corresponding predicted output vector $f_\theta$, for time step $i$ for the $j$-th trajectory.
- $\| \cdot \|^2$ denotes the squared Euclidean norm.

Minimizing data loss constrains PINN to approximate a mapping to replicate observed trajectory patterns in the training data.

*First-Order Physics Loss*. The physics loss component, $\mathcal{L}_{\text{phy}}(\theta)$, is the cornerstone of the PINN framework. It enforces physical plausibility by penalizing deviations from a simplified kinematic model of vessel motion. Specifically, we use a *forward Euler approximation* (a first-order finite difference method) of the kinematic equations to estimate the expected change in latitude and longitude over a discrete time step $\Delta t$. This expected change is then compared against the change predicted by the neural network. To estimate the expected displacement, we consider the input parameters at time $t$: $v_t$ (SOG), $\psi_t$ (COG), $a_t$ (acceleration), and $\dot{\psi}_t$ (COG rate). To improve the first-order approximation, we incorporate a midpoint $\psi_{mid,rad}$ derived from the *Taylor series expansion*:

$$\psi\left(t + \frac{\Delta t}{2}\right) \approx \psi(t) + \dot{\psi}(t) \cdot \frac{\Delta t}{2} \tag{4}$$

Approximating the rate of change of $\dot{\psi}(t)$ and converting angles to radians, we obtain the midpoint:

$$\psi_{mid,rad} = \psi_{t,rad} + \frac{1}{2} \cdot \dot{\psi}_{t,rad} \cdot \Delta t \tag{5}$$

where $\psi_{t,rad} = \psi_t \cdot \frac{\pi}{180}$ and $\dot{\psi}_{t,rad} = \dot{\psi}_t \cdot \frac{\pi}{180}$ are the COG and COG rate in radians, respectively.

Using the midpoint $\psi_{mid,rad}$, the *expected* changes in latitude and longitude, $\Delta x_{\text{expected}}$ and $\Delta y_{\text{expected}}$, can be computed using either a small-angle approximation or a great-circle approximation to Earth's curvature; both approaches were adopted in our evaluation. The *small-angle approximation* offers computational efficiency, suitable for short prediction horizons, and is formulated as:

$$
\begin{aligned}
\Delta x_{\text{expected}} \approx \big( v_t \cdot \cos(\psi_{\text{mid, rad}}) \\
+ \frac{1}{2} \cdot a_t \cdot \Delta t \cdot \cos(\psi_{\text{mid, rad}}) \big) \cdot \text{factor} \\
\Delta y_{\text{expected}} \approx \big( v_t \cdot \sin(\psi_{\text{mid, rad}}) \\
+ \frac{1}{2} \cdot a_t \cdot \Delta t \cdot \sin(\psi_{\text{mid, rad}}) \big) \cdot \frac{\text{factor}}{\cos(x_{t,\text{rad}})}
\end{aligned}
\tag{6}
$$

where factor $= \frac{\Delta t}{R} \cdot \frac{180}{\pi}$, $R$ is Earth's radius, $\Delta x_{expected}$ and $\Delta y_{expected}$ refer to known x and y coordinated displacements, as observed in the data.

Conversely, for enhanced accuracy, particularly over longer prediction horizons, the *great-circle approximation* is employed, formulated as:

$$
\begin{aligned}
\Delta x_{expected} = \arcsin \big( \sin(x_{t,rad}) \cdot \cos(d) \\
+ \cos(x_{t,rad}) \cdot \sin(d) \cdot \cos(\psi_{mid,rad}) \big) - x_{t,rad} \\
\Delta y_{expected} = \arctan 2 \big( \sin(\psi_{mid,rad}) \cdot \sin(d) \cdot \cos(x_{t,rad}), \\
\cos(d) - \sin(x_{t,rad}) \cdot \sin(x_{t,rad}) \big)
\end{aligned}
\tag{7}
$$

where the angular distance $d$, accounting for Earth's radius $R$ (approximately 6,371,000 meters), is computed as:

$$d = \frac{\big( v_t + \frac{1}{2} \cdot a_t \cdot \Delta t \big) \cdot \Delta t}{R} \tag{8}$$

The *predicted x and y displacements* in latitude and longitude from the neural network are computed as finite differences:

$$
\begin{aligned}
\Delta x_{\text{pred},t+i} = \hat{x}_{t+i+1} - \hat{x}_{t+i} \\
\Delta y_{\text{pred},t+i} = \hat{y}_{t+i+1} - \hat{y}_{t+i}
\end{aligned}
\tag{9}
$$

The mismatch between these predicted and expected changes is then expressed by the *finite difference physics residuals*:

$$
\begin{aligned}
r_{x,t+i} = \Delta x_{\text{pred},t+i} - \Delta x_{\text{expected},t+i} \\
r_{y,t+i} = \Delta y_{\text{pred},t+i} - \Delta y_{\text{expected},t+i}
\end{aligned}
\tag{10}
$$

Finally, the physics loss $\mathcal{L}_{\text{phy}}(\theta)$ is computed by averaging the sum of squared residuals over all time steps within the prediction horizon and all trajectories in the batch:

$$\mathcal{L}_{\text{phy}}(\theta) = \frac{1}{N \cdot H} \sum_{i=1}^{N} \sum_{j=1}^{H} \big( r_{x,i,t+j}^2 + r_{y,i,t+j}^2 \big) \tag{11}$$

***Second-Order Physics Loss.*** Despite using midpoint approximations, the physics loss based on the first-order forward Euler method limits its ability to capture dynamics within each time-step ($\Delta t$). This limitation leads to discretization errors, especially for longer time steps, as it only considers derivatives at the beginning of the interval. To address this limitation, we introduce a second-order physics loss using Heun's method (a type of Runge-Kutta 2), which offers a balance between computational simplicity and improved accuracy over the first-order approach, making it particularly suitable for modeling maritime trajectory dynamics.

Heun's method improves on Euler by incorporating derivative information at both the beginning and an estimated endpoint of the interval through a predictor-corrector approach. First, the *predictor step* uses the forward Euler method to obtain an initial estimate of the state at $t + \Delta t$, denoted with a superscript $P$:

$$
\begin{aligned}
x_{t+\Delta t}^{P} = x_t + \dot{x}_t \cdot \Delta t \\
y_{t+\Delta t}^{P} = y_t + \dot{y}_t \cdot \Delta t
\end{aligned}
\tag{12}
$$

where $\dot{x}_t = \frac{dx}{dt}\big|_t$ and $\dot{y}_t = \frac{dy}{dt}\big|_t$ are the derivatives evaluated using the kinematic model with the vessel's state at time $t$.

Second, the *corrector step* refines this estimate by computing the derivatives again, using the kinematic model evaluated at the *predicted* state at $t + \Delta t$. These derivatives are denoted as follows.

$$\dot{x}_{t+\Delta t}^{P} = \frac{dx}{dt}\bigg|_{x_{t+\Delta t}^{P}, y_{t+\Delta t}^{P}} , \qquad \dot{y}_{t+\Delta t}^{P} = \frac{dy}{dt}\bigg|_{x_{t+\Delta t}^{P}, y_{t+\Delta t}^{P}}$$

Heun's method then averages the derivatives from the beginning ($t$) and the predicted end ($t+\Delta t$) to compute the final *expected* change in state over the interval $\Delta t$:

$$
\begin{aligned}
\Delta x_{\text{expected}} = \frac{1}{2}(\dot{x}_t + \dot{x}_{t+\Delta t}^{P}) \cdot \Delta t \\
\Delta y_{\text{expected}} = \frac{1}{2}(\dot{y}_t + \dot{y}_{t+\Delta t}^{P}) \cdot \Delta t
\end{aligned}
\tag{13}
$$

These computed increments, $\Delta x_{\text{expected}}$ and $\Delta y_{\text{expected}}$, represent the kinematically *expected changes* for the second-order physics loss $\mathcal{L}_{\text{phy}}(\theta)$. This loss function then quantifies the discrepancies between these *expected displacements* and the corresponding *predicted displacements* by the neural network, $\Delta x_{\text{pred},t+i}$ and $\Delta y_{\text{pred},t+i}$ (Eqs. 9). The resulting *residuals*, $r_{x,t+i}$ and $r_{y,t+i}$, computed as the difference between predicted and expected displacements (Eqs. 10), are squared and averaged, yielding the final second-order physics loss value (Eq. 11). Minimizing this loss constrains the PINN to produce trajectories that not only fit the data but also strongly adhere to the second-order kinematic approximations provided by Heun's method.

***Total Loss Function.*** The total loss function, $\mathcal{L}_{\text{total}}(\theta)$, minimized during the PINN training process, is a weighted combination of the data and physics loss:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{data}}(\theta) + \lambda \cdot \mathcal{L}_{\text{phy}}(\theta) \tag{14}$$

Here, $\lambda \geq 0$ is a non-negative hyperparameter controlling the relative importance of the physics loss in the total loss. By adjusting $\lambda$, the model can balance its adherence to observed data with its compliance to the kinematic model.

For consistent scaling in model training and physical fidelity in loss computation, input kinematic features ($x_t, y_t, v_t, \psi_t, a_t, \dot{\psi}_t$) are

first normalized to $[0, 1]$ and then denormalized back to their original units to compute physics-informed residuals. To balance their contribution in the loss, these residuals—quantifying discrepancies between predicted and expected displacements—are normalized by the latitude and longitude ranges in the training set. We further assume that $v_t$, $\psi_t$, $a_t$, and $\dot{\psi}_t$ remain constant at their last observed values throughout the prediction horizon $H$, providing a practical basis for enforcing kinematic constraints without future input.

## 4 EVALUATION

This section details the experimental setup and presents a comprehensive performance analysis of our PINN framework, implemented in Python 3 using the Keras/TensorFlow libraries.

### 4.1 Dataset and Preprocessing

For evaluation, we selected two distinct AIS datasets from different maritime regions, sourced from AISViz/MERIDIAN [23]. The first dataset originates from the Arctic region (LON -95 to -75, LAT 55 to 66), which is characterized by sparse vessel traffic due to a navigation season limited to 4 months annually. In contrast, the second dataset is derived from the Strait of Georgia (LON -128 to -122, LAT 48 to 51), a region known for its high density of vessel traffic, but trajectories with frequent turns and speed alterations. We collected the Arctic dataset, which spans two years (2022-2023) and a dataset of six months (January - June, 2023) from the Strait of Georgia.

For each region, AIS messages were extracted, retaining the following attributes: MMSI, timestamp, latitude, longitude, Speed Over Ground (SOG), Course Over Ground (COG), and ship type. These extracted AIS messages were then filtered based on ship type and vessel identification number (MMSI). Subsequently, vessel trajectories were extracted for each region by grouping the filtered AIS messages according to their MMSI. Finally, we prepared the AIS dataset for evaluation by applying the following preprocessing steps to each extracted trajectory.

***Noise Filtering***. Trajectories with invalid MMSI identifiers and AIS messages with duplicate timestamps were removed. To focus on moving vessels, AIS messages with SOG below 0.5 knots (indicating anchored vessels) were excluded. Trajectories were retained only if they exhibited a minimum length of 300 data points. Finally, COG values were wrapped to the range [0, 360) degrees to address issues of non-north GPS settings.

***Time-based Trip Segmentation***. Vessel trajectories, spanning long periods and regions, often include multiple trips and stationary periods. To isolate individual trips, we segmented trajectories if the time gap between consecutive points exceeded 60 minutes.

***Cubic Hermite Interpolation***. Trajectory segments are interpolated into 2-minute intervals using *Cubic Hermite splines*. This method uses position and derivative (slope) information at each data point for interpolation, ensuring smooth transitions without overshooting or oscillations and preserving the trajectory's shape. This makes it well-suited for capturing realistic vessel maneuvers, including abrupt direction and speed changes.

***Kinematic Feature Derivation***. Vessel SOG ($v_t$) and COG ($\psi_t$) can be influenced by external factors such as wind, ocean currents, and weather conditions. Therefore, deriving acceleration ($a_t$) and COG rate ($\dot{\psi}_t$) from these values is crucial to capture changes in vessel mobility patterns and modes. Acceleration ($a_t$) represents the rate of change of SOG ($v_t$) over time, while COG rate ($\dot{\psi}_t$) is the rate of change of COG ($\psi_t$). In our dataset, SOG values were converted from *knots* to $m/s$. Acceleration ($m/s^2$) and COG rate ($degrees/s$) were then derived using finite differences as follows:

$$a_i = \frac{v_i - v_{i-1}}{t_i - t_{i-1}}, \quad \dot{\psi}_i = \frac{\psi_i - \psi_{i-1}}{t_i - t_{i-1}} \tag{15}$$

After preprocessing the trajectories of cargo and tanker vessels from both regions, we kept only 3 hour segments to ensure consistent input and output sequences for model training and testing. Key statistics of the resulting datasets reveal the following: for the Strait of Georgia, we have 4, 315 cargo segments (819, 265 data points) and 773 tanker segments (161, 331 data points); for the Arctic region, the dataset comprises 277 cargo segments (143, 399 data points) and 216 tanker segments (76, 675 data points). Their spatial distributions are visualized in Figures 2 and 3, respectively.
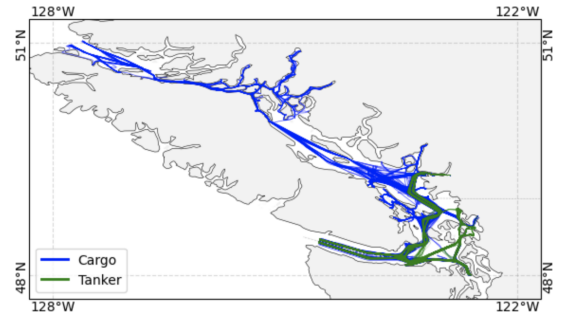


**Figure 2: Dense vessel traffic in the Strait of Georgia, depicted by Cargo and Tanker trajectories.**
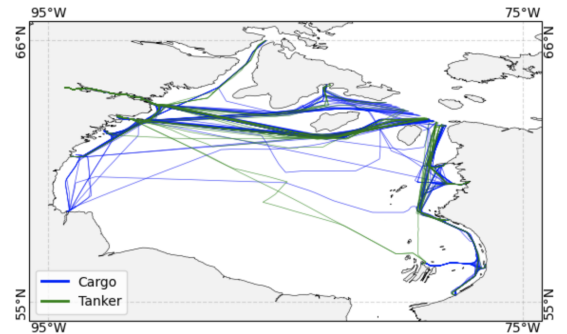


**Figure 3: Cargo and Tanker trajectories in the Arctic, highlighting sparse maritime traffic.**

### 4.2 Evaluation Metrics

To evaluate the accuracy of the prediction models, we employed both *classical regression* metrics, such as Mean Absolute Error (MAE) and Mean Squared Error (MSE), and *state-of-the-art* trajectory prediction metrics, such as Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE and FDE directly quantify

spatial accuracy, which is critical for navigational decision making in maritime contexts.

MSE averages the squared differences between predicted $(\hat{y}_i)$ and actual $(y_i)$ locations. Although MSE is computationally straightforward and emphasizes larger errors, it is highly sensitive to outliers. MAE averages the absolute differences, treating all errors equally and exhibiting greater robustness to outliers. The formulas for MAE and MSE are given in Eq. 16, where $n$ denotes sample size.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (16)$$

ADE represents the average distance between predicted $(\hat{x}_i, \hat{y}_i)$ and actual $(x_i, y_i)$ locations over all time steps, providing a comprehensive measure of trajectory accuracy. Lower ADE values indicate better overall prediction. FDE, in contrast, measures only the distance between the final predicted $(\hat{x}_n, \hat{y}_n)$ and actual $(x_n, y_n)$ locations. Lower FDE values suggest better accuracy in predicting the trajectory endpoint and reduced error accumulation over time. The formulas for ADE and FDE are given in Eq. 17, where $n$ is the trajectory length.

$$\text{ADE} = \frac{1}{n}\sum_{i=1}^{n}H_d\big((x_i, y_i), (\hat{y}_i, \hat{y}_i)\big)$$
$$\text{FDE} = H_d\big((x_n, y_n), (\hat{x}_n, \hat{y}_n)\big) \qquad (17)$$

where $H_d$ represents the *Haversine* distance between two points, calculated as:

$$H_d = 2 \cdot R \cdot \arcsin\sqrt{\sin^2\left(\frac{\Delta x}{2}\right) + \cos(x_1) \cdot \cos(x_2) \cdot \sin^2\left(\frac{\Delta y}{2}\right)} \quad (18)$$

Here, $R$ represents the Earth's radius, and $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$, where $x$ and $y$ denote latitude and longitude in radians, respectively. Distance errors are computed in meters in this evaluation.

## 4.3 Setting Model I/O and Hyperparameters

***DL Model Architectures***. We selected various state-of-the-art Deep Learning (DL) models that have been widely used by researchers in trajectory prediction and sequence modeling. This selection consists of a range of architectures, including Recurrent Neural Network (RNN)-based models (LSTMs and GRUs), Convolutional Neural Networks (CNNs), Temporal Convolutional Networks (TCNs), Hybrid architectures (ConvLSTMs), and Transformers.

***Data Preparation for Model I/O***. To prepare the dataset for DL models, we begin by performing feature-wise normalization using Min-Max scaling and subsequently apply a sliding-window approach. For each trajectory, we extracted the input sequences $X$ and the corresponding target sequences $Y$ by sliding a window of length $W_{\text{in}} + W_{\text{out}}$. Given a trajectory and denoting $t$ as the last time step of the input window, the input sequence $X$ comprises $W_{\text{in}}$ consecutive feature vectors ending at time step $t$:

$$X_t = \{\mathbf{x}_{t-W_{\text{in}}+1}, \mathbf{x}_{t-W_{\text{in}}+2}, \ldots, \mathbf{x}_t\}$$

The corresponding target sequence $Y$, representing the prediction horizon $W_{\text{out}}$, consists of $W_{\text{out}}$ output vectors starting from time step $t + 1$:

$$Y_t = \{\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+W_{\text{out}}}\}$$

Applying this sliding window generated input-target pairs $(X_t, Y_t)$ from all trajectories for training, validation, and testing. For each vessel type, 10% of the trajectories were assigned to the test set. Of the remaining trajectories, 80% for training and 20% for validation.

***Hyperparameter Settings***. The physics loss weight, $\lambda$, is a crucial hyperparameter for PINNs. To assess its influence, we performed experiments on $\lambda$ values ranging from 0.0001 to 1.0. All other hyperparameters were kept consistent across models and datasets. All models were trained with Adam optimizer (initial learning rate 0.001) for up to 50 epochs and with a batch size of 32. The models were optimized for MSE, but performance was monitored using both MSE and MAE. Hidden layers used ReLU activations, while the output layers employed linear activations. Training was regularized using Early Stopping and the ReduceLROnPlateau scheduler, based on validation performance.

We used an encoder-only Transformer configured with 2 layers, 128 model dimensions, 4 attention heads, and 256 feed-forward dimensions, along with dropout and default positional encoding. For TCN, we used residual blocks with increasing dilation rates $\{2^i\}_{i=0}^4$. For other models, we used 64 hidden units for 1-layer and 64/32 units for 2-layer setups.

## 4.4 Performance Analysis

To comprehensively assess efficacy, we evaluated our PINN integrated models against baselines without physical constraints, based on the following criteria:

(1) **Model Complexity:** We varied the complexity of DL models to investigate their ability to balance data-driven accuracy with physics constraints. **Basic** refers to simpler configurations (*e.g.*, single-layer LSTMs, GRUs, CNNs, or fewer TCN residual blocks), while **Complex** refers to models with deeper or wider layers, or additional residual blocks.

(2) **Waterway Complexity:** We considered waterway complexity in the performance analysis, as areas such as ports or waypoints are more complex than open seas. Two cases were considered: **Case 1** (all sliding windows) includes both dynamic and open-seas, whereas **Case 2** (beginning, middle, and end windows) focuses on dynamic areas.

(3) **Prediction Horizon:** We varied the lengths of the prediction horizon to assess the models' robustness against temporal uncertainty and error accumulation.

(4) **Score Selection:** Instead of relying on the lowest ADE/FDE scores for assessment, we selected these scores based on physics plausibility, robust validation performance, and synergistic convergence, as monitored through training/validation data loss and physics loss curves across epochs.

(5) **Numerical Approximation:** We assessed the performance based on the order of numerical approximations.

***Arctic Dataset***. Table 1 presents the mean ADE and FDE for various DL models, evaluated with and without PINN regularization across two levels of model complexity. The results consistently show that incorporating PINN reduces both ADE (up to 32%) and FDE (up to 27%) across nearly all model types and complexity levels. However, as model complexity increases, the performance gains for architectures like ConvLSTM and TCN do not improve significantly,

likely due to their higher data requirements for optimal convergence. Nonetheless, these complex models with PINN still achieve significantly lower errors compared to their non-PINN counterparts and are close to their basic implementations, suggesting that PINN regularization enables effective model training even with relatively small datasets.

**Table 1: Model performance by level of model complexity.**

| | Models | No PINN | | First–Order PINN | |
|---|---|---|---|---|---|
| | | ADE | FDE | ADE | FDE |
| **Basic** | LSTM | 2584 ± 840 | 4354 ± 1508 | **2296** ± 679 | 3950 ± 1469 |
| | GRU | 2769 ± 793 | 4913 ± 1383 | **2477** ± 741 | 4447 ± 1288 |
| | CNN | 3385 ± 1036 | 5409 ± 1818 | **3336** ± 841 | 5406 ± 1300 |
| | ConvLSTM | 1613 ± 529 | 2615 ± 873 | **1384** ± 454 | 2215 ± 871 |
| | TCN | 2036 ± 751 | **3428** ± 1386 | **1811** ± 643 | 3444 ± 1334 |
| **Complex** | LSTM | 2176 ± 789 | 3483 ± 1532 | **2081** ± 655 | 3474 ± 1322 |
| | GRU | 2542 ± 833 | 4330 ± 1288 | **2218** ± 635 | 3898 ± 1430 |
| | CNN | 2687 ± 939 | 4570 ± 1530 | **2552** ± 698 | 4285 ± 1262 |
| | ConvLSTM | 1961 ± 737 | 3010 ± 1432 | **1596** ± 503 | 2383 ± 872 |
| | TCN | 2652 ± 1703 | 4479 ± 2367 | **1800** ± 520 | 3232 ± 1085 |

*Dataset = Arctic, Vessel = Tanker, $W_{in}$ = 30min, $W_{out}$ = 30min, $\Delta t$ = 2min*

Table 2 shows the model performance trends as prediction horizons increase based on best performing models, ConvLSTM and TCN. It is evident that the First–Order PINN consistently reduces ADE across all prediction horizons. The percentage decrease compared to No PINN models ranges from approximately 11% to over 37% for *basic* models, while it ranges from 15% to 33% for *complex* models. Overall, the percentage reduction in ADE through the integration of PINN is consistently positive on all horizons and models, demonstrating its robust performance enhancement.

**Table 2: Model performance by prediction horizon length.**

*Dataset = Arctic, Vessel = Tanker, $W_{in}$ = 30m, $W_{out}$ = 10/20/30m, $\Delta t$ = 2min*

| | Models | No PINN (ADE) | | | First Order PINN (ADE) | | |
|---|---|---|---|---|---|---|---|
| | | 10m | 20m | 30m | 10m | 20m | 30m |
| **Basic** | ConvLSTM | 1047 | 1258 | 1613 | 895 | 1091 | 1384 |
| | TCN | 1259 | 1565 | 2036 | 788 | 1254 | 1811 |
| **Complex** | ConvLSTM | 1395 | 1616 | 1961 | 1189 | 1318 | 1596 |
| | TCN | 1384 | 1997 | 2652 | 1018 | 1330 | 1800 |

As stated before, vessel maneuvers differ significantly between dynamic areas (*e.g.*, ports and waypoints), while course and speed adjustments are less frequent in open seas. Therefore, averaging performance across entire trajectories (*Case 1*) can obscure the nuanced benefits of physics-informed models, particularly in areas like the Arctic (see Figure 3). To specifically assess the impact of physics approximations, we also strategically selected test windows (*Case 2*) to capture varying navigation contexts. Table 3 reveals that prediction errors (ADE and FDE) are higher in Case 2 than in Case 1, indicating greater complexity in the selected test windows. Notably, the integration of First–Order PINN consistently reduces errors in both cases, highlighting its effectiveness with smaller datasets regardless of navigational complexity.

***Strait of Georgia Dataset***. As a dense and narrow waterway, the Georgia Strait exhibits more complex and dynamic vessel movement patterns compared to the Arctic. Although it is not a large dataset, the 6-month Georgia dataset contains more trajectories than 2 years of Arctic data. As shown in Table 4, overall average and

**Table 3: Model performance by complexity of water area.**

*Dataset = Arctic, Vessel = Tanker, $W_{in}$ = 30m, $W_{out}$ = 30m, $\Delta t$ = 2min*

| | Basic Models | No PINN | | First-Order PINN | |
|---|---|---|---|---|---|
| | | ADE | FDE | ADE | FDE |
| **Case 1** | ConvLSTM | 1613 ± 529 | 2615 ± 873 | 1384 ± 454 | 2215 ± 871 |
| | TCN | 2036 ± 751 | **3428** ± 1386 | 1811 ± 643 | 3444 ± 1334 |
| **Case 2** | ConvLSTM | 2169 ± 776 | 3248 ± 1202 | 1716 ± 596 | 2674 ± 1035 |
| | TCN | 2409 ± 861 | 4057 ± 1556 | 1999 ± 886 | 3512 ± 1734 |

final displacement errors were significantly reduced compared to the Arctic dataset (see Table 4) as models converged better with more data. Unlike the Arctic, both open seas and port areas are geometrically complex in this dataset. Consequently, we computed errors for all sliding windows (Case 1), where the PINN approach yielded ADE error reductions of up to approximately 12% and FDE 15%, highlighting its subtle yet crucial benefits.

**Table 4: Displacement errors - No PINN *vs.* First-Order PINN**

*Dataset = Georgia, Vessel = Tanker, $W_{in}$ = 30min, $W_{out}$ = 30min, $\Delta t$ = 2min*

| Complex Models | No PINN | | First–Order PINN | |
|---|---|---|---|---|
| | ADE | FDE | ADE | FDE |
| LSTM | 740 ± 219 | 1394 ± 515 | **722** ± 204 | 1358 ± 461 |
| GRU | 805 ± 222 | 1566 ± 529 | **799** ± 224 | 1549 ± 500 |
| CNN | 1318 ± 480 | 2288 ± 838 | **1162** ± 378 | 2014 ± 682 |
| ConvLSTM | **637** ± 197 | 1177 ± 447 | 654 ± 203 | 1237 ± 427 |
| TCN | 800 ± 233 | 1596 ± 518 | **784** ± 216 | 1369 ± 494 |

Furthermore, MAE and MSE were computed for both latitude and longitude (see Table 5), which show that longitude errors are consistently larger than latitude errors, indicating that east-west motion is harder to predict. The first-order PINN significantly reduced longitude errors (MAE by 2–10% and MSE by up to 23%), especially for the CNN model. This indicates that the kinematic constraint mainly reduces the noisier longitude errors, whereas the already small latitude errors remain largely unchanged and contribute little to the overall prediction.

**Table 5: PINN Impact on Spatial Dimensions.**

*Dataset = Georgia, Vessel = Tanker, $W_{in}$ = 30min, $W_{out}$ = 30min, $\Delta t$ = 2min*

| Complex Models | No PINN | | | | First–Order PINN | | | |
|---|---|---|---|---|---|---|---|---|
| | MAE | | MSE | | MAE | | MSE | |
| | LAT | LON | LAT | LON | LAT | LON | LAT | LON |
| LSTM | 0.0041 | 0.0066 | 0.00005 | 0.00012 | 0.0041 | **0.0064** | 0.00005 | 0.00012 |
| GRU | 0.0046 | 0.0070 | 0.00006 | 0.00014 | 0.0046 | **0.0069** | 0.00006 | 0.00014 |
| CNN | 0.0074 | 0.0115 | 0.00016 | 0.00035 | 0.0065 | **0.0104** | 0.00011 | 0.00027 |
| ConvLSTM | 0.0036 | **0.0057** | 0.00004 | 0.00010 | 0.0036 | 0.0059 | 0.00004 | 0.00010 |
| TCN | 0.0044 | 0.0072 | 0.00006 | 0.00016 | 0.0043 | **0.0070** | 0.00006 | 0.00015 |

***First–Order vs. Second–Order Physics***. We also analyzed whether adding second-order dynamics to PINN improves predictive accuracy compared to a first-order approximation, specifically on the smaller Arctic dataset. As shown in Table 6, switching from a first-order to a second-order PINN yields marginal gains in ADE and FDE across models (*e.g.*, ConvLSTM reduced ADE slightly from 1596 to 1558). This subtle difference is expected, as the Arctic dataset reflects less dynamic conditions, featuring predominantly open-sea navigation with low traffic density. Also, predicting over a short horizon (≤ 30 min) with small time steps (2 min) means higher-order effects have limited time to accumulate and significantly impact the predictions. Moreover, the first-order PINN has already

achieved a significant improvement. For TCN, the second-order PINN does not yield further improvement, but when considering the complex area (Case 2), it reduces ADE and FDE to 1927 and 3408 respectively, from first-order errors (see Table 3).

**Table 6: Model performance by order of approximations.**

| $Dataset = Arctic, Vessel = Tanker, W_{in} = 30min, W_{out} = 30min, \Delta t = 2min$ | | | | |
|---|---|---|---|---|
| **Complex Models** | **First−Order PINN** | | **Second−Order PINN** | |
| | **ADE** | **FDE** | **ADE** | **FDE** |
| LSTM | $2081 \pm 655$ | $3474 \pm 1322$ | **1973** $\pm$ 507 | **3408** $\pm$ 1252 |
| GRU | $2218 \pm 635$ | $3898 \pm 1430$ | **2198** $\pm$ 587 | **3841** $\pm$ 1379 |
| CNN | $2552 \pm 698$ | $4285 \pm 1262$ | **2222** $\pm$ 715 | **3750** $\pm$ 1346 |
| ConvLSTM | $1596 \pm 503$ | $2383 \pm 872$ | **1558** $\pm$ 536 | $2319 \pm 900$ |
| TCN | **1800** $\pm$ 520 | **3232** $\pm$ 1085 | $1823 \pm 610$ | $3465 \pm 1332$ |

***Transformer***. To further assess PINN efficacy, we also evaluated an encoder-only Transformer, whose capacity to model long-range dependencies suits complex sequential tasks like trajectory prediction. Given the Transformer's inherent requirement for substantial training data to achieve convergence, we conducted experiments using the Georgia dataset. As presented in Table 7, both ADE and FDE errors exhibited reductions when switching from No PINN settings to First-Order PINN in both cases.

**Table 7: Model performance by complexity of water area and order of numerical approximations.**

| $Dataset = Georgia, Vessel = Tanker, W_{in} = 30min, W_{out} = 30min, \Delta t = 2min$ | | | | | | |
|---|---|---|---|---|---|---|
| **Transformer** | **No PINN** | | **First−Order PINN** | | **Second−Order PINN** | |
| | **ADE** | **FDE** | **ADE** | **FDE** | **ADE** | **FDE** |
| **Case 1** | 605 | 1046 | **554** | **992** | 573 | 1023 |
| **Case 2** | 691 | 1112 | **629** | **1087** | 645 | 1092 |

Although Second−order PINN did not yield further improvements in terms of ADE and FDE magnitudes, it is noteworthy that the standard deviation of displacement errors decreased, as evidenced by the reduced spread in the box plot of displacement errors shown in Figure 4.
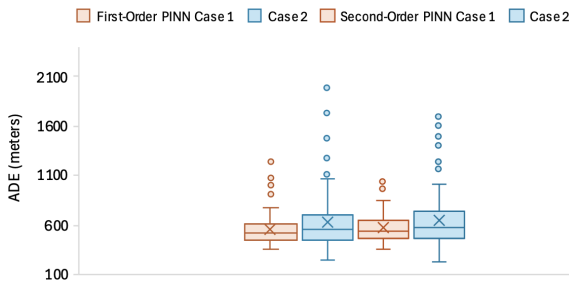


**Figure 4: Transformer - First Order *versus* Second Order.**

***PINNs Complexity***. PINN training typically requires more time than standard supervised learning due to the additional physics loss computation. As shown in Figure 5, using our finite difference approach, the training time increases moderately for the first-order PINN and significantly for the second-order PINN compared to the non-PINN baseline. Moreover, tuning the weight parameter $\lambda$ is necessary to achieve optimal convergence. However, the proposed finite difference approach is faster and consumes less memory compared to standard PINNs that rely on automatic differentiation,

which requires storing and computing intermediate operations and their derivatives via a computational graph.
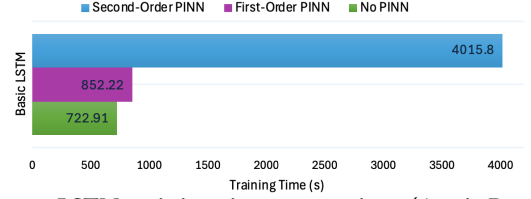


**Figure 5: LSTM training time comparison (Arctic Dataset).**

***Sample Prediction Results***. Figures 6 and 7 show predicted tracks from the Arctic and Georgia regions, respectively. These tracks are generated by combining predictions across all sliding windows using the best performing models for each region: ConvLSTM for the Arctic and Transformer for Georgia. As observed in Figure 6, the track predicted by the Non-PINN ConvLSTM significantly deviates from the ground truth. The ADE of tracks predicted by ConvLSTM is 1417m for Non-PINN and 982m for First-Order PINN. In Figure 7, the track predicted by the Non-PINN Transformer shows many spikes, indicating notable deviations from the reference line. The ADE in this case 535m and 463m for Non-PINN and First-Order PINN, respectively. This also proves that even when a comparatively good amount of data is available, a PINN model can enforce kinematic constraints and improve prediction accuracy.
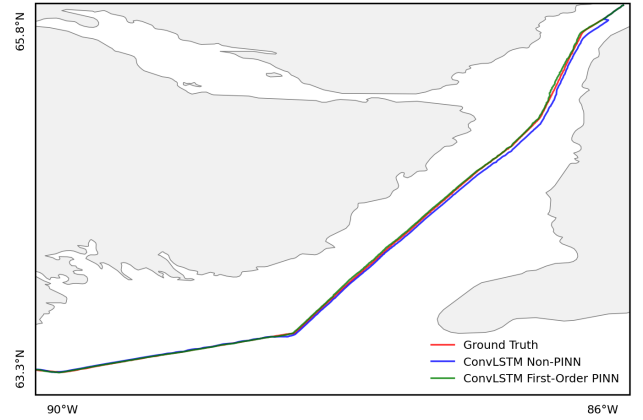


**Figure 6: Arctic - Observed *vs*. ConvLSTM Predicted Tracks.**

## 5 CONCLUSION AND FUTURE WORK

Recent DL-based approaches for vessel trajectory prediction are data-intensive and lack explicit integration of vessel motion dynamics. Consequently, their prediction accuracy suffers due to sudden vessel maneuvers or in response to complex weather conditions. To address these limitations, we proposed a PINN-based approach, leveraging first- and second-order numerical approximations implemented via forward finite differences. Our experimental results demonstrate that this approach provides a valuable trade-off, effectively combining data-driven learning with physical consistency, while exhibiting improved performance even with limited data.

While training PINNs can be computationally demanding and requires careful hyperparameter tuning, this approach presents a
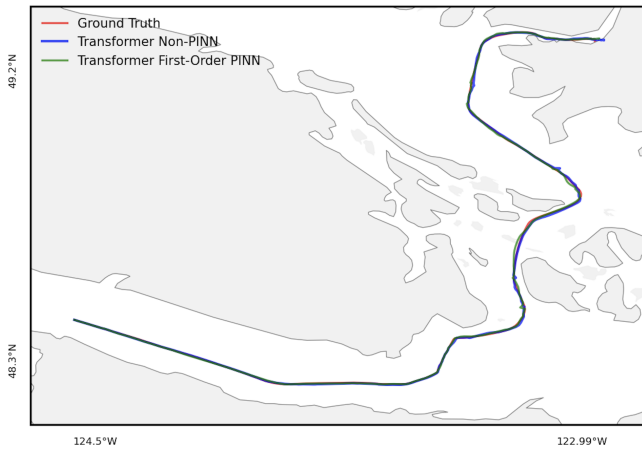
**Figure 7: Strait of Georgia - Observed *vs.* Transformer Predicted Tracks.**

compelling direction towards more robust, accurate, and generalizable prediction models. To further advance this research, future work will integrate higher-order numerical approximations to improve prediction accuracy in complex maritime scenarios and over longer prediction horizons. The security and safety of maritime navigation are also inherently dependent on inter-vessel interactions. Consequently, our ongoing goal is to incorporate spatial relationships among vessels directly into our PINN framework, further ensuring physical plausibility in complex multi-vessel scenarios.

## Acknowledgments

## References

[1] Md M. Alam, G. Spadon, M. Etemad, L. Torgo, and E. Milios. 2024. Enhancing short-term vessel trajectory prediction with clustering for heterogeneous and multi-modal movement patterns. *Ocean Engineering* 308 (2024), 118303.

[2] P. Antão, S. Sun, A.P. Teixeira, and C. Guedes Soares. 2023. Quantitative assessment of ship collision risk influencing factors from worldwide accident and fleet data. *Reliability Engineering & System Safety* 234 (6 2023), 109166.

[3] Hao Chen, Daqi Zhu, and Mingzhi Chen. 2025. PGL: A short-time model for ship trajectory prediction. *Journal of Navigation* (1 2025), 1–17.

[4] J. Chen, J. Zhang, H. Chen, Y. Zhao, and H. Wang. 2023. A TDV attention-based BiGRU network for AIS-based vessel trajectory prediction. *Iscience* 26, 4 (2023).

[5] S. Cuomo, V. Schiano di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. 2022. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *Journal of Scientific Computing* 92 (1 2022), 88.

[6] Martha Dais Ferreira and Jessica N. A. Campbell. 2025. A novel RNN architecture to improve the precision of ship trajectory predictions. *Applied Artificial Intelligence* 39 (12 2025), 2459465. Issue 1. doi:10.1080/08839514.2025.2459465

[7] Zhiyuan Guo, Huimin Qiang, and Xiaodong Peng. 2025. Vessel Trajectory Prediction Using Vessel Influence Long Short-Term Memory with Uncertainty Estimation. *Journal of Marine Science and Engineering* 13, 2 (2025), 353.

[8] Zhiyuan Guo, Huimin Qiang, and Xiaodong Peng. 2025. Vessel Trajectory Prediction Using Vessel Influence Long Short-Term Memory with Uncertainty Estimation. *Journal of Marine Science and Engineering* 13 (2 2025), 353.

[9] J. Jiang, Y. Zuo, Y. Xiao, W. Zhang, and T. Li. 2024. STMGF-Net: a spatiotemporal multi-graph fusion network for vessel trajectory forecasting in intelligent maritime navigation. *IEEE Trans. on Intelligent Transportation Systems* (2024).

[10] Xiao Lang, Da Wu, and Wengang Mao. 2024. Physics-informed machine learning models for ship speed prediction. *Expert Sys. with App.* 238 (3 2024), 121877.

[11] Huanhuan Li, Hang Jiao, and Zaili Yang. 2023. Ship trajectory prediction based on machine learning and deep learning: A systematic review and methods analysis. *Engineering Applications of Artificial Intelligence* 126 (11 2023), 107062.

[12] Siyuan Lin, Yufei Jiang, Feng Hong, Lixiang Xu, Haiguang Huang, and Bin Wang. 2025. HDFormer: A transformer-based model for fishing vessel trajectory prediction via multi-source data fusion. *Ocean Engineering* 320 (2025), 120309.

[13] Zu Lin, Weiqi Yue, Jie Huang, and Jian Wan. 2023. Ship trajectory prediction based on the TTCN-attention-GRU model. *Electronics* 12, 12 (2023), 2556.

[14] Michail Mathioudakis, Christos Papandreou, Theodoros Stouraitis, Vicky Margari, Antonios Nikitakis, Stavros Paschalakis, Konstantinos Kyriakopoulos, and Kostas J. Spyrou. 2025. Towards Real-World Validation of a Physics-Based Ship Motion Prediction Model. *arXiv* (1 2025).

[15] Duong Nguyen and Ronan Fablet. 2024. A transformer network with sparse augmented data representation and cross entropy loss for ais-based vessel trajectory prediction. *IEEE Access* 12 (2024), 21596–21609.

[16] ORBCOMM. 2016. AIS: The Largest Maritime Paradigm Shift in Over a Century. https://blog.orbcomm.com/satellite-ais-the-largest-maritime-paradigm-shift-in-over-a-century/. Last Accessed: June, 2024.

[17] Christos Papandreou, Michail Mathioudakis, Theodoros Stouraitis, Petros Iatropoulos, Antonios Nikitakis, Stavros Paschalakis, and Konstantinos Kyriakopoulos. 2025. Interpretable Data-Driven Ship Dynamics Model: Enhancing Physics-Based Motion Prediction with Parameter Optimization. *arXiv* (2 2025).

[18] C. Ray, A.L. Jousselme, C. Iphar, M. Zocholl, E. Camossi, and R. Dréo. 2020. Mobility data: A perspective from the maritime domain. *Big Data Analytics for Time-Critical Mobility Forecasting: From Raw Data to Trajectory-Oriented Mobility Analytics in the Aviation and Maritime Domains* (2020), 3–31.

[19] Yuyol Shin, Namwoo Kim, Hyeyeong Lee, Soh Young In, Mark Hansen, and Yoonjin Yoon. 2024. Deep learning framework for vessel trajectory prediction using auxiliary tasks and convolutional networks. *Engineering Applications of Artificial Intelligence* 132 (2024), 107936.

[20] I. Slaughter, J. L. Charla, M. Siderius, and J. Lipor. 2024. Vessel trajectory prediction with recurrent neural networks: An evaluation of datasets, features, and architectures. *Journal of Ocean Engineering and Science* (2024).

[21] SOLAS. 2004. Consolidated text of the International Convention of Safety of Life at Sea, 1974, and its Protocol of 1988: articles, annexes and certificates. *IMO* (2004).

[22] Ruixin Song, Gabriel Spadon, Ronald Pelot, Stan Matwin, and Amilcar Soares. 2024. Enhancing global maritime traffic network forecasting with gravity-inspired deep learning models. *Scientific Reports* 14 (7 2024), 16665. Issue 1.

[23] Gabriel Spadon, Jay Kumar, Jinkun Chen, Matthew Smith, Casey Hilliard, Sarah Vela, Romina Gehrmann, Claudio DiBacco, Stan Matwin, and Ronald Pelot. 2024. Maritime tracking data analysis and integration with AISdb. *SoftwareX* 28 (2024).

[24] Gabriel Spadon, Jay Kumar, Derek Eden, Josh van Berkel, Tom Foster, Amilcar Soares, Ronan Fablet, Stan Matwin, and Ronald Pelot. 2024. Multi-path long-term vessel trajectories forecasting with probabilistic feature fusion for problem shifting. *Ocean Engineering* 312 (2024), 119138.

[25] Yongfeng Suo, Zhengnian Ding, and Tao Zhang. 2024. The Mamba Model: A Novel Approach for Predicting Ship Trajectories. *Journal of Marine Science and Engineering* 12 (8 2024), 1321. Issue 8. doi:10.3390/jmse12081321

[26] J. D. Toscano, V. Oommen, A. J. Varghese, Z. Zou, N. A. Daryakenari, C. Wu, and G. E. Karniadakis. 2024. From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning. *Machine Learning for Computational Science and Engineering* 1 (10 2024), 1–43. Issue 1.

[27] Zheng Wan and Jihong Chen. 2018. Human errors are behind most oil-tanker spills. *Nature* 560 (8 2018), 161–163. Issue 7717. doi:10.1038/d41586-018-05852-0

[28] Xinyun Wu, Jiafei Chen, Caiquan XIONG, Donghua Liu, Xiang Wan, and Zexi Chen. 2024. Vessel Trajectory Prediction Method Based on the Time Series Data Fusion Model. *Promet-Traffic&Transportation* 36, 6 (2024), 1160–1175.

[29] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh. 2019. Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey. *IEEE Trans. on Intelligent Transportation Systems* 21, 5 (2019), 1796–1825.

[30] Dong Yang, Kaisen Yang, Yuxu Lu, Maohan Liang, and Congcong Zhao. 2024. Graph-driven multi-vessel long-term trajectories prediction for route planning under complex waters. *Ocean Engineering* 313 (2024), 119511.

[31] Xiaocai Zhang, Xiuju Fu, Zhe Xiao, Haiyan Xu, and Zheng Qin. 2022. Vessel Trajectory Prediction in Maritime Transportation: Current Approaches and Beyond. *IEEE Transactions on Intelligent Transportation Systems* 23 (11 2022), 19980–19998. Issue 11.

[32] Xiliang Zhang, Jin Liu, Chengcheng Chen, Lai Wei, Zhongdai Wu, and Wenjuan Dai. 2025. Goal-driven long-term marine vessel trajectory prediction with a memory-enhanced network. *Expert Systems with Applications* 263 (2025), 125715.

[33] Zhiwen Zhang, Wei Yuan, Zipei Fan, Xuan Song, and Ryosuke Shibasaki. 2025. AISFuser: Encoding Maritime Graphical Representations With Temporal Attribute Modeling for Vessel Trajectory Prediction. *IEEE Transactions on Knowledge and Data Engineering* 37 (4 2025), 1571–1584. Issue 4.

[34] Licheng Zhao, Yi Zuo, Wenjun Zhang, Tieshan Li, and C.L. Philip Chen. 2025. SNIINet: Trajectory prediction using ship navigation information interaction-aware neural network. *Ocean Engineering* 321 (3 2025), 120368.

[35] Yifeng Zhao, Zhiqiang Hu, Weifeng Du, Lingbo Geng, and Yi Yang. 2024. Research on Modeling Method of Autonomous Underwater Vehicle Based on a Physics-Informed Neural Network. *Journal of Marine Science and Eng.* 12 (5 2024), 801.

[36] Dimitrios Zissis, Elias K Xidias, and Dimitrios Lekkas. 2016. Real-time vessel behavior prediction. *Evolving Systems* 7 (2016), 29–40.