

Article

Motion-Decoupled Dual-Stream Representation Learning for AIS-Based Vessel Trajectory Prediction

Chiming Wang ^{1,*}, Dongke Zheng ¹, Yiyang Zhou ¹, Rongjiong Wu ¹, Shunzhi Zhu ¹, Qin Nie ¹, Zhenjun Li ² and Bingkun Wu ³

¹ School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China; 17779465020@163.com (D.Z.); 15211682168@163.com (Y.Z.); 18815972252@163.com (R.W.); szzhu@xmut.edu.cn (S.Z.); nieqin@xmut.edu.cn (Q.N.)

² Fujian Province Shipbuilding Industry Group Co., Ltd., Fuzhou 350015, China; lzj@fsigc.com

³ Zhongshu Xinke, Zhongshu (Xiamen) Information Technology Co., Ltd., Xiamen 361006, China; wubk@crowddigital.cn

* Correspondence: algorithm_007@163.com

Abstract

Automatic Identification System (AIS)-based vessel trajectory prediction is essential for maritime traffic management and navigation safety. Existing deep learning methods typically model vessel motion within a unified temporal representation space, which may entangle long-term navigation trends with local maneuvering behaviors. However, vessel trajectories inherently exhibit heterogeneous dynamics, including steady route evolution and non-stationary maneuver perturbations. To address this issue, this paper proposes MD-EDTCNFormer, a motion-decoupled dual-stream framework for vessel trajectory prediction. A Global Navigation Dynamics Encoder is designed to capture dominant route-level temporal evolution from raw AIS sequences, while a Residual Maneuver Dynamics Encoder explicitly models maneuver-related local perturbations through state transition residual representations. In addition, a state-adaptive motion aggregation mechanism is introduced to dynamically balance global navigation dependencies and local maneuver-aware dynamics under different navigation states. Depthwise separable temporal convolution and efficient channel attention are further integrated to suppress redundant temporal-channel coupling and emphasize dynamically dominant motion cues. Experiments on a real-world AIS dataset from the Zhoushan coastal area demonstrate the effectiveness of the proposed framework under coastal traffic conditions, and show improvements in prediction accuracy and trajectory stability compared with representative baseline methods.

Keywords: vessel trajectory prediction; automatic identification system; temporal convolutional network; transformer; efficient channel attention



Academic Editor: Rodolfo T. Gonçalves

Received: 1 June 2026

Revised: 10 July 2026

Accepted: 23 July 2026

Published: 24 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](#) license.

1. Introduction

With the rapid growth of maritime transportation, vessel trajectory prediction has become increasingly important for traffic management, collision avoidance, route planning, and autonomous navigation [1–4]. The widespread use of the AIS provides large-scale spatiotemporal vessel data, promoting the development of data-driven trajectory prediction methods [5,6]. Early studies mainly relied on statistical models and physical motion assumptions [7], such as Kalman filtering and autoregressive methods [8,9]. Although computationally efficient, these approaches have limited capability in modeling nonlinear vessel motion under complex maritime conditions. Recently, deep learning methods,

including RNNs, LSTMs, GRUs, TCNs, and Transformers, have shown strong performance in sequential trajectory modeling [10–13].

However, most existing methods model vessel motion within a unified temporal representation space, implicitly assuming homogeneous motion dynamics. In practice, vessel trajectories contain heterogeneous temporal patterns [14]. On the one hand, route-level navigation usually exhibits smooth and long-term temporal continuity. On the other hand, maneuver behaviors, such as turning, acceleration, and collision avoidance, are often abrupt and non-stationary. Encoding these different dynamics within a unified representation may lead to representation conflict, where long-range navigation trends interfere with maneuver-sensitive local motion modeling.

In addition, AIS sequences often contain strong temporal continuity and redundant inter-channel correlations. Conventional temporal convolutions jointly model temporal dependencies and channel interactions, which may introduce redundant feature coupling and weaken sensitivity to critical motion variations [15]. Moreover, the importance of motion variables changes under different navigation states. For example, route-level dynamics dominate during steady navigation, while maneuver-related residual dynamics become more important during turning or speed changes. Existing methods generally lack adaptive mechanisms to balance these heterogeneous motion dependencies.

To address these issues, this paper proposes a motion-decoupled dual-stream framework for AIS-based vessel trajectory prediction. It should be noted that the proposed method is intended for short-term trajectory prediction in monitored waters, rather than centralized vessel traffic control. Its outputs serve as auxiliary information for vessel traffic services, port authorities, coastal monitoring systems, and risk-assessment modules, while final navigation decisions remain with vessel operators and relevant maritime authorities. The proposed method explicitly disentangles global navigation dynamics and local maneuver dynamics through separate temporal representations. A Global Navigation Dynamics Encoder is used to capture route-level temporal evolution from raw AIS sequences, while a Residual Maneuver Dynamics Encoder models maneuver-related perturbations using state transition residuals. By separating steady navigation trends from non-stationary maneuver dynamics, the proposed framework improves heterogeneous motion representation.

Furthermore, depthwise separable temporal convolution is introduced to decouple temporal dependency extraction from channel interaction modeling, reducing redundant temporal-channel coupling [16]. Efficient channel attention is employed to adaptively emphasize dynamically important motion cues [17]. A state-adaptive motion aggregation mechanism is further designed to balance global navigation dependencies and local maneuver dynamics under different navigation states. The main contributions of this paper are summarized as follows:

1. A motion-decoupled temporal representation framework is proposed for vessel trajectory prediction, which reformulates trajectory forecasting as a heterogeneous motion dynamics modeling problem.
2. A dual-stream temporal encoding architecture is designed to separately model dominant navigation evolution and maneuver-related residual dynamics through explicit motion disentanglement.
3. Depthwise separable temporal convolution and efficient channel attention are integrated to suppress redundant temporal-channel coupling and adaptively emphasize dynamically dominant motion cues.
4. A state-adaptive motion aggregation mechanism is developed to dynamically balance global navigation dependencies and local maneuver-aware dynamics for robust vessel trajectory prediction.

2. Related Work

2.1. Traditional and Machine Learning-Based Methods

Early vessel trajectory prediction studies mainly relied on physical models, statistical inference, and traditional machine learning algorithms [18,19]. These methods usually describe vessel motion using explicit state-transition rules, probabilistic models, or handcrafted trajectory features, and therefore have relatively good interpretability.

Kalman filter-based methods have been used for vessel state estimation and short-term trajectory prediction. Perera et al. [7] adopted an extended Kalman filter to recursively estimate vessel motion states. Markov-based methods regard vessel movement as a probabilistic state-transition process [20]. Liu et al. [21] proposed a K-order multivariate Markov chain model for long-term trajectory prediction. Gaussian-process-based methods have also been applied to trajectory uncertainty estimation. In addition, Nie et al. [22] introduced an EMD-SVR model to improve short-term ship motion prediction for non-stationary time-series signals.

Traditional machine learning methods further exploit historical AIS data for trajectory or destination prediction. Borkowski [23] combined multi-sensor navigation data fusion with adaptive neural networks. Tu et al. [24] developed a large-scale AIS-based path prediction framework to handle low data quality, irregular sampling, and trajectory diversity. Zhang et al. [25] and Ma et al. [26] used random-forest-based methods for vessel destination prediction and behavior modeling.

Although these methods provide useful solutions for AIS-based trajectory prediction, they are often constrained by handcrafted features, motion assumptions, or trajectory-matching rules. Therefore, their ability to capture nonlinear route evolution, long-term temporal dependencies, and local maneuvering behaviors in complex maritime environments remains limited.

2.2. Deep Learning-Based Methods

With the development of deep learning, data-driven sequential models have been widely applied to vessel trajectory prediction [27,28]. Compared with traditional methods, deep learning models can automatically learn temporal representations from historical AIS sequences and better capture nonlinear vessel motion.

Recurrent neural networks and their variants, such as LSTM, GRU, BiLSTM, and Seq2Seq models, are commonly used for AIS-based trajectory prediction [29–32]. Tang et al. [29] used LSTM to learn vessel motion sequences from AIS data. Liu et al. [33] proposed an H-LSTM model that combines data-driven prediction with Kalman filter-based motion estimation, while Liu et al. [34] introduced QSD-LSTM by embedding quaternion ship-domain knowledge to model vessel interactions in complex encounter scenarios.

Beyond recurrent structures, convolutional and hybrid deep models have been developed to enhance local trajectory feature extraction. Wu et al. [35] proposed a ConvLSTM-Seq2Seq model for joint spatial-temporal feature learning. Lin et al. [36] combined temporal convolution, attention, and GRU to capture multi-scale temporal dependencies. More recent studies further explored hybrid structures, including SeqLSTM-U-Net [37], multi-scale convolutional attention LSTM [38], intention-aware dual-joint attention networks [39], and bidirectional state-space models with squeeze-and-excitation attention [40].

Transformer-based models have also received increasing attention because self-attention can directly model dependencies among historical time steps. Jiang et al. [41] proposed TRFM-LS by incorporating LSTM into the Transformer framework. Xue et al. [13] developed G-Trans, a GRU-based Transformer model for spatiotemporal feature extraction. Qiang et al. [42] introduced motion constraints and dynamic-aware attention into

Transformer-based prediction, while Cao et al. [43] designed SGAformer to capture short-term disturbances and long-term trajectory trends.

Graph-based models have been increasingly adopted to capture vessel interactions in multi-ship trajectory prediction. Zhao et al. [44] proposed GAT-LSTM, which combines graph attention with LSTM to learn spatial dependencies and temporal dynamics from AIS trajectories. Zhang et al. [45] developed ASTGCN with a StemGNN-derived correlation matrix to enhance inter-feature dependency modeling. Wu et al. [46] introduced GL-STGCNN, where ship interaction adjacency matrices are learned dynamically and further corrected by model predictive control. More recently, Gong et al. [47] proposed STGTP, a spatiotemporal graph Transformer that integrates temporal attention, graph-based spatial interaction modeling, and probabilistic uncertainty representation for more robust trajectory prediction.

Although deep learning methods have advanced vessel trajectory prediction, most existing approaches still encode raw AIS variables within a unified representation, implicitly assuming homogeneous motion dynamics. This limits their ability to disentangle route-level navigation trends from maneuver-level variations and to effectively exploit explicit kinematic cues. These limitations highlight the need for motion-decoupled modeling to better capture heterogeneous vessel motion dynamics.

3. Methodology

3.1. Problem Formulation and Motion-State Decomposition

This study formulates vessel trajectory prediction from historical AIS records, which describe vessel position, motion, and time-varying spatiotemporal evolution. The model uses raw AIS states and residual motion states to predict short-term future positions. It does not require destination, voyage plan, or ownership information, and the route-level evolution refers to observed motion trends rather than predetermined routes.

Given a historical AIS sequence with length T , the task is to predict the vessel positions in the next L time steps. At time step t , the AIS navigation state is defined as

$$\mathbf{s}_t^g = [\text{lat}_t, \text{lon}_t, \text{soc}_t, \text{cog}_t] \quad (1)$$

where lat_t , lon_t , soc_t , and cog_t denote latitude, longitude, speed over ground, and course over ground, respectively. The superscript g is used to indicate that this state is mainly used to describe the global navigation trend of the vessel. The historical navigation-state sequence is written as

$$\mathbf{S}^g = [\mathbf{s}_1^g, \mathbf{s}_2^g, \dots, \mathbf{s}_T^g] \quad (2)$$

the historical navigation-state sequence $\mathbf{S}^g$ is organized as a tensor $\mathbf{S}^g \in \mathbb{R}^{B \times T \times 4}$, where B denotes the batch size, T denotes the historical observation length, and the four channels correspond to latitude, longitude, speed over ground, and course over ground, respectively.

Vessel motion usually contains both a relatively stable navigation trend and local motion variations. The former reflects the route-level movement pattern, while the latter is related to maneuvering behaviors such as turning, acceleration, and deceleration. To distinguish these two types of motion information, the vessel state is described from two perspectives:

$$\mathbf{S}_t = \{\mathbf{s}_t^g, \mathbf{s}_t^r\} \quad (3)$$

where $\mathbf{s}_t^g$ represents the navigation-state component obtained from raw AIS records, and $\mathbf{s}_t^r$ represents the residual motion component derived from state changes between adjacent time steps.

The residual motion component is constructed from position and velocity variations. The position residuals are defined as

$$\begin{cases} \Delta lat_t = lat_t - lat_{t-1} \\ \Delta lon_t = lon_t - lon_{t-1} \end{cases} \quad (4)$$

Since the course over ground (COG) in AIS records is measured clockwise from north, with 0° indicating north and 90° indicating east, SOG and COG are first converted into eastward and northward Cartesian velocity components:

$$\begin{cases} v_t^x = sog_t \sin\left(cog_t \frac{\pi}{180}\right) \\ v_t^y = sog_t \cos\left(cog_t \frac{\pi}{180}\right) \end{cases} \quad (5)$$

where v_t^x and v_t^y denote the eastward and northward velocity components, respectively. This Cartesian representation avoids directly differencing raw course angles across the $0^\circ/360^\circ$ boundary, because course values close to this boundary are mapped to nearby velocity vectors. The corresponding velocity-component residuals are then calculated as

$$\begin{cases} \Delta v_t^x = v_t^x - v_{t-1}^x \\ \Delta v_t^y = v_t^y - v_{t-1}^y \end{cases} \quad (6)$$

where Δv_t^x and Δv_t^y denote the changes in eastward and northward velocity, respectively. The residual motion state at time step t is then defined as

$$\mathbf{s}_t^r = [\Delta lat_t, \Delta lon_t, \Delta v_t^x, \Delta v_t^y] \quad (7)$$

Compared with the raw AIS state, $\mathbf{s}_t^r$ describes how the vessel state changes over time. It therefore provides direct information about local maneuvering behavior and complements the navigation trend represented by $\mathbf{s}_t^g$. The residual-state sequence is written as

$$\mathbf{S}^r = [\mathbf{s}_1^r, \mathbf{s}_2^r, \dots, \mathbf{s}_T^r] \quad (8)$$

where $\mathbf{S}^r \in \mathbb{R}^{B \times T \times 4}$.

The ground-truth future trajectory is defined as

$$\mathbf{Y} = [\mathbf{y}_{T+1}, \mathbf{y}_{T+2}, \dots, \mathbf{y}_{T+L}], \quad \mathbf{y}_k = [lat_k, lon_k] \quad (9)$$

accordingly, the ground-truth future trajectory is organized as $\mathbf{Y} \in \mathbb{R}^{B \times L \times 2}$, where B is the batch size, L is the prediction horizon, and the last dimension corresponds to latitude and longitude.

The trajectory prediction problem is formulated as

$$\hat{\mathbf{Y}} = f(\mathbf{S}^g, \mathbf{S}^r) \quad (10)$$

where $\hat{\mathbf{Y}}$ is the predicted future trajectory. In this formulation, $\mathbf{S}^g$ provides the raw navigation-state sequence, and $\mathbf{S}^r$ provides the residual motion sequence. The two sequences are later encoded by different branches to model route-level evolution and local maneuvering information separately.

3.2. Overall Framework

The proposed MD-EDTCNFormer follows a motion-decoupled representation scheme, and its overall architecture is shown in Figure 1. Instead of encoding all motion patterns in a single latent space, the framework separates vessel trajectory modeling into two

complementary streams: a global navigation dynamics stream and a maneuver residual dynamics stream.

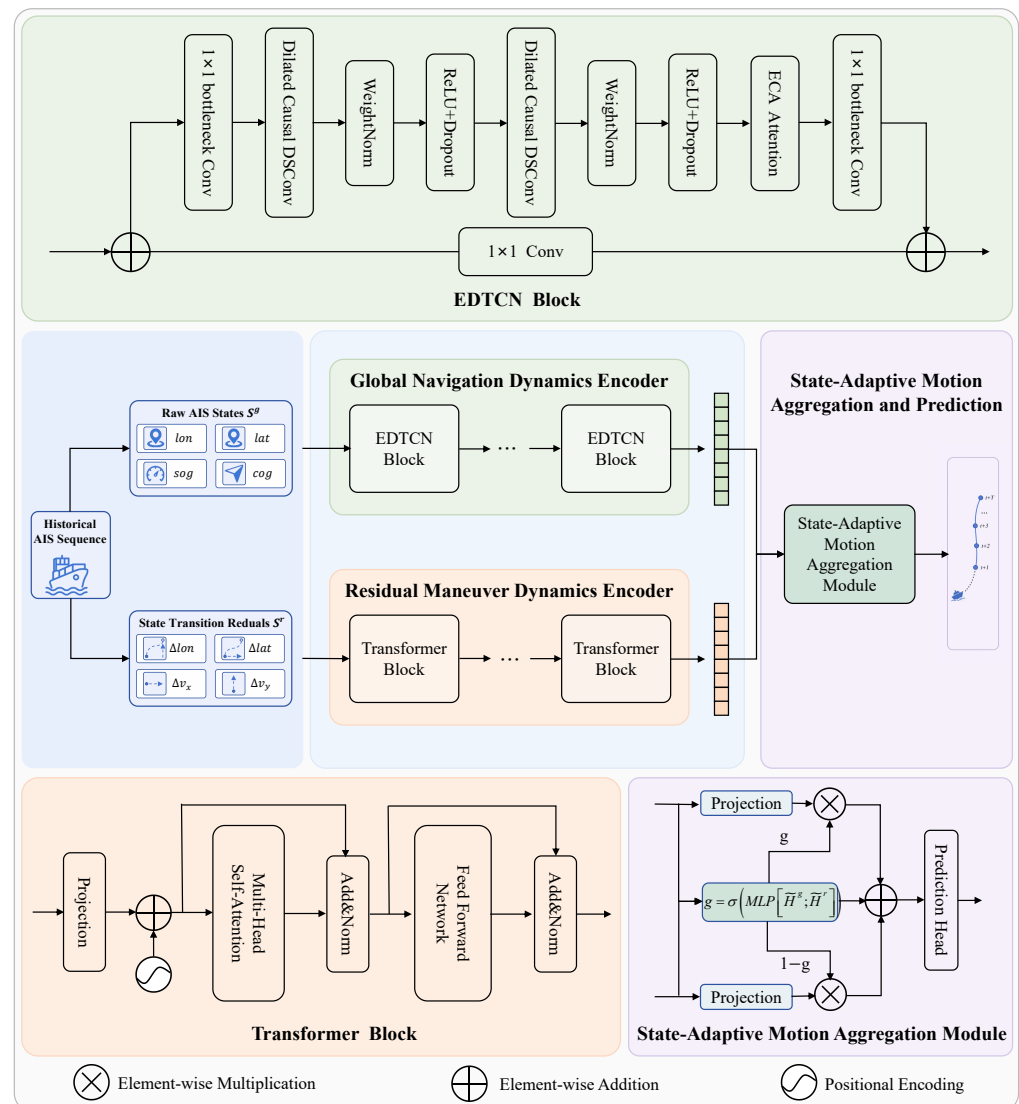


Figure 1. Overall architecture of the proposed MD-EDTCNFormer for motion-decoupled AIS-based vessel trajectory prediction.

The global navigation dynamics stream takes the raw AIS sequence S^g as input and models route-level temporal evolution. This stream focuses on stable and continuous motion patterns, such as main sailing direction and long-term route tendency. The maneuver residual dynamics stream takes the state transition residual sequence S^r as input and models local motion deviations caused by turning, acceleration, deceleration, and other non-stationary behaviors.

The overall framework consists of three parts. First, an EDTCN encoder is used to extract multi-scale route-level temporal representations from the raw AIS sequence. Depth-wise separable temporal convolution is adopted to reduce redundant temporal-channel coupling, and efficient channel attention is used to adjust channel responses according to motion saliency. Second, a Transformer encoder is applied to the residual motion sequence to model temporal dependencies among maneuver-related state transitions. Third, a state-adaptive aggregation module combines the two streams by learning their relative

contributions under different navigation states. The two encoded representations are denoted as

$$\begin{aligned}\mathbf{H}^g &= F_{\text{GNDE}}(\mathbf{S}^g), \\ \mathbf{H}^r &= F_{\text{RMDE}}(\mathbf{S}^r)\end{aligned}\quad (11)$$

In this formulation, $F_{\text{GNDE}}(\cdot)$ and $F_{\text{RMDE}}(\cdot)$ denote the Global Navigation Dynamics Encoder and the Residual Maneuver Dynamics Encoder, respectively. They produce compact motion representations $\mathbf{H}^g \in \mathbb{R}^{B \times d_g}$ and $\mathbf{H}^r \in \mathbb{R}^{B \times d_r}$, where d_g and d_r are the hidden dimensions of the two streams. Specifically, $\mathbf{H}^g$ captures route-level navigation dynamics, while $\mathbf{H}^r$ models residual maneuver dynamics. The two representations are then fused for multi-step trajectory prediction.

Compared with single-stream prediction models, the proposed framework avoids forcing heterogeneous motion patterns into one representation space. The EDTCN stream preserves the continuity of dominant navigation evolution, while the Transformer stream supplies residual motion information that is sensitive to local maneuvers.

3.3. Global Navigation Dynamics Encoder

Route-level vessel motion usually exhibits temporal continuity across multiple scales. Short-term variations reflect local position changes, while long-range dependencies describe the evolution of the sailing route. Temporal convolutional networks are suitable for this setting because causal convolution preserves chronological order and dilated convolution expands the receptive field without introducing future information. The causal dilated convolution structure of TCN is illustrated in Figure 2.

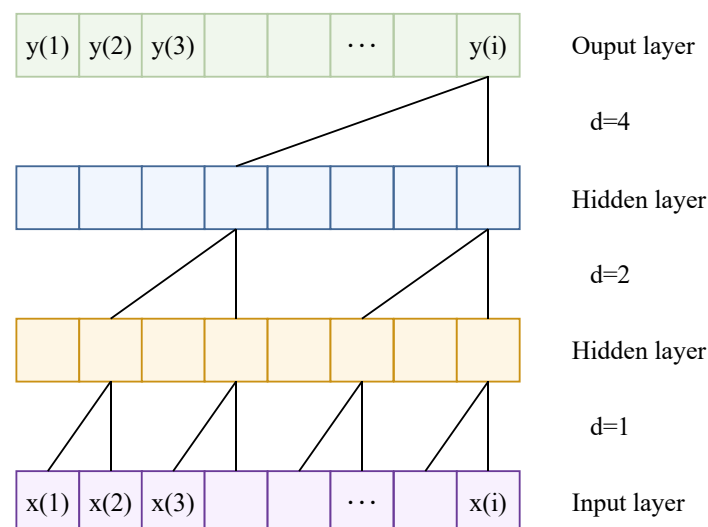


Figure 2. Schematic diagram of the causal dilated convolution structure in TCN.

For the i -th temporal block, the dilation rate is defined as

$$d_i = 2^i, \quad i = 0, 1, \dots, L_d - 1 \quad (12)$$

where L_d is the number of stacked temporal blocks. With increasing dilation rates, shallow layers focus on local motion continuity, whereas deeper layers capture longer-range route evolution.

A standard temporal convolution jointly processes temporal dependencies and channel interactions. For AIS sequences, such full coupling may introduce redundant cross-channel computation because latitude, longitude, speed, and course contain strong correlations during steady navigation. To reduce this redundancy, depthwise separable dilated causal

convolution is used in each Light Temporal Block. The trimming length caused by causal padding is calculated as

$$p_i = (k - 1)d_i \quad (13)$$

where k is the convolution kernel size. The Chomp operation removes redundant padded positions and maintains causality. The basic structure of depthwise separable convolution is shown in Figure 3.

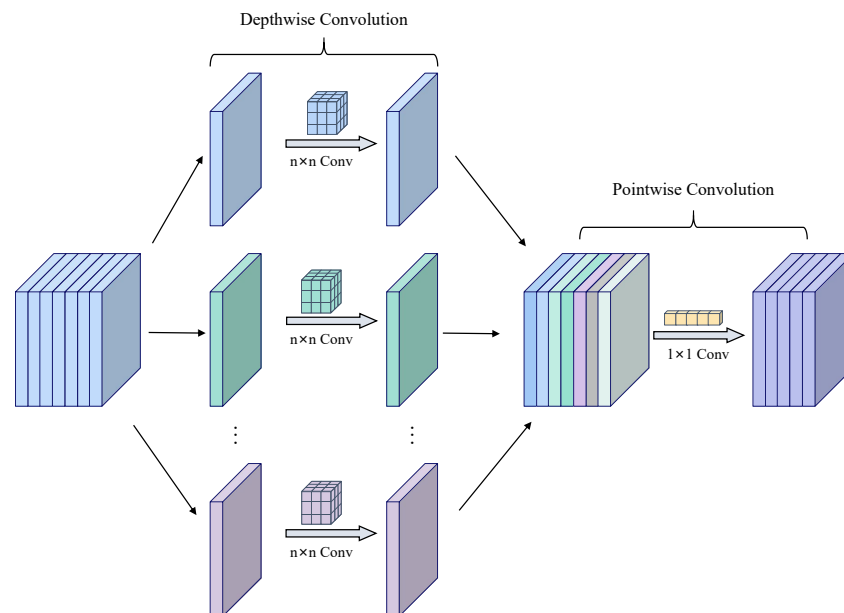


Figure 3. Schematic diagram of depthwise separable convolution.

In the global navigation stream, $\mathbf{S}^g \in \mathbb{R}^{B \times T \times 4}$ is first transposed into the temporal-convolution format $\mathbb{R}^{B \times 4 \times T}$. Inside the i -th EDTCN block, the input of the j -th depthwise separable causal convolution is denoted as $\mathbf{H}_{j-1}^{(i)} \in \mathbb{R}^{B \times d_b \times T}$, where d_b is the bottleneck channel dimension. The depthwise separable temporal convolution is written as

$$\begin{aligned} \tilde{\mathbf{H}}_j^{(i)} &= \text{DWConv}_{k,d_i}(\mathbf{H}_{j-1}^{(i)}), \\ \hat{\mathbf{H}}_j^{(i)} &= \text{Chomp}_{p_i}(\tilde{\mathbf{H}}_j^{(i)}), \\ \mathbf{H}_j^{(i)} &= \text{PWConv}(\hat{\mathbf{H}}_j^{(i)}), \\ \mathbf{Z}_j^{(i)} &= \text{Dropout}(\text{ReLU}(\mathbf{H}_j^{(i)})) \end{aligned} \quad (14)$$

$\text{DWConv}_{k,d_i}(\cdot)$ performs channel-wise temporal filtering with kernel size k and dilation rate d_i , $\text{Chomp}_{p_i}(\cdot)$ removes the redundant positions at the end of the temporal dimension, while $\text{PWConv}(\cdot)$ uses a 1×1 convolution to fuse information across channels. In this way, temporal dependency extraction and channel interaction modeling are separated. The encoder can retain multi-scale temporal modeling capacity while reducing redundant channel coupling.

After temporal filtering, efficient channel attention is introduced to adjust the response of motion-related channels. Different AIS variables may have different importance under different navigation states. For example, position continuity may dominate during steady sailing, while velocity variation becomes more informative during maneuvering. The ECA module provides a lightweight channel recalibration mechanism without dimensionality reduction. The structure of the ECA module is shown in Figure 4.

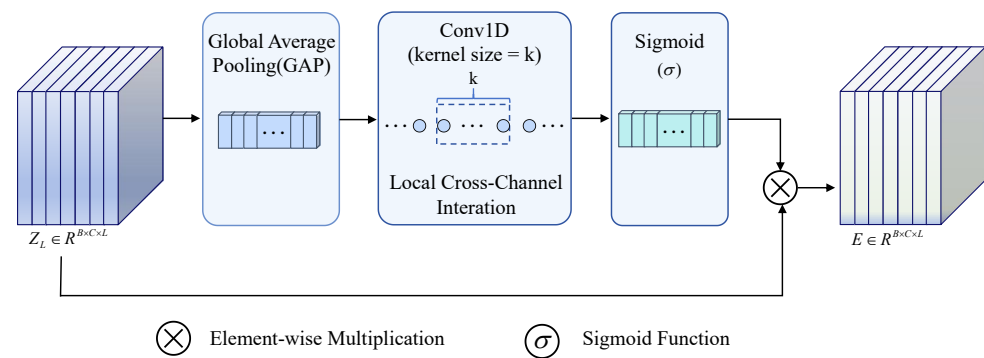


Figure 4. Schematic diagram of the ECA channel attention mechanism.

Because causal padding followed by the Chomp operation preserves the temporal length, the output of the last convolutional layer in the i -th EDTCN block is denoted as $\mathbf{Z}_L^{(i)} \in \mathbb{R}^{B \times d_b \times T}$. The channel attention weights are then computed from this feature as follows:

$$\begin{aligned} \mathbf{s}^{(i)} &= \text{GAP}(\mathbf{Z}_L^{(i)}), \\ \mathbf{a}^{(i)} &= \sigma(\text{Conv1D}(\mathbf{s}^{(i)})), \\ \mathbf{E}^{(i)} &= \mathbf{a}^{(i)} \odot \mathbf{Z}_L^{(i)} \end{aligned} \quad (15)$$

where $\text{GAP}(\cdot)$ denotes global average pooling, $\text{Conv1D}(\cdot)$ captures local channel interactions, $\sigma(\cdot)$ is the Sigmoid function, and $\odot$ denotes element-wise multiplication. The attention weights $\mathbf{a}^{(i)}$ modulate channel-wise motion saliency and suppress weak responses. The ECA module recalibrates channel responses while preserving the tensor shape, producing the enhanced feature $\mathbf{E}^{(i)} \in \mathbb{R}^{B \times d_b \times T}$. The attention weights $\mathbf{a}^{(i)}$ emphasize informative motion-related channels and suppress weak responses.

After the 1×1 convolutional projection and residual connection, the output tensor of the block is represented as $\mathbb{R}^{B \times d_g \times T}$. The feature at the last time step is selected as the global navigation representation, denoted as $\mathbf{H}^g \in \mathbb{R}^{B \times d_g}$. This representation encodes the dominant route-level evolution of the historical AIS sequence.

3.4. Residual Maneuver Dynamics Encoder

The raw AIS sequence contains both route-level trends and local maneuver variations. However, maneuvering behaviors are often short-lived and non-stationary, and may be weakened when all variables are encoded directly in one temporal representation. The residual sequence $\mathbf{S}^r$ is therefore processed by a separate maneuver dynamics stream.

In the residual maneuver stream, the residual sequence $\mathbf{S}^r \in \mathbb{R}^{B \times T \times 4}$ is first mapped into the Transformer hidden space through a linear projection:

$$\mathbf{Z}_p = \mathbf{S}^r \mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}} \quad (16)$$

where $\mathbf{W}_{\text{in}}$ and $\mathbf{b}_{\text{in}}$ denote the weight matrix and bias of the input projection layer, respectively. To preserve the temporal order of the residual sequence, a learnable positional embedding is added to the projected representation:

$$\mathbf{Z} = \mathbf{Z}_p + \mathbf{P} \quad (17)$$

where $\mathbf{P}$ denotes the learnable positional embedding matrix. After the input projection and positional embedding, the Transformer input is obtained as $\mathbf{Z} \in \mathbb{R}^{B \times T \times d_r}$.

Maneuver residuals may exhibit abrupt temporal changes. For example, a turning behavior may depend not only on the immediately preceding step but also on several

earlier velocity and heading variations. Self-attention is used to model such dependencies across the residual sequence. The scaled dot-product attention is defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \quad (18)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ denote the query, key, and value matrices, respectively, and d_k is the dimension of the key vectors. In the Transformer encoder, these tensors are represented as $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times T \times d_r}$.

For the h -th attention head, the projected query, key, and value matrices are used to compute one temporal dependency representation:

$$\text{head}_h = \text{Attention}\left(\mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V\right) \quad (19)$$

where $\mathbf{W}_h^Q$, $\mathbf{W}_h^K$, and $\mathbf{W}_h^V$ are learnable projection matrices for the h -th head. With n_h attention heads, the dimension of each head is $d_h = d_r/n_h$, and the output of each head satisfies $\text{head}_h \in \mathbb{R}^{B \times T \times d_h}$.

Each attention head learns temporal relations from a different representation subspace. The structure of multi-head attention is shown in Figure 5.

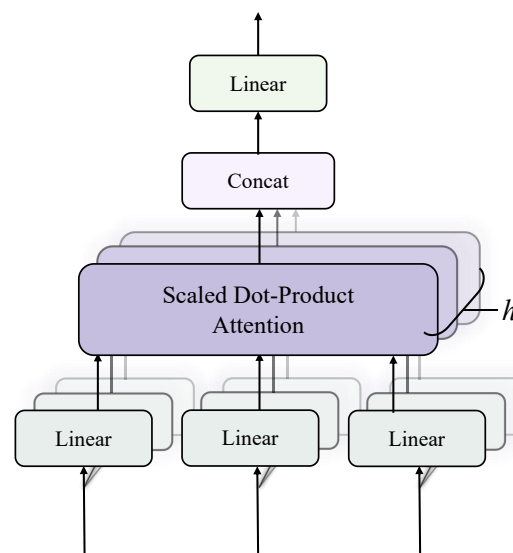


Figure 5. Schematic diagram of the multi-head attention mechanism.

The outputs of all heads are then concatenated and linearly projected to form the multi-head attention output:

$$\text{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_{n_h})\mathbf{W}^O \quad (20)$$

where $\mathbf{W}^O$ is the output projection matrix. After concatenation and output projection, the multi-head attention output remains in $\mathbb{R}^{B \times T \times d_r}$.

The attention output is further processed through residual connections, layer normalization, and a feed-forward network. Finally, the hidden state at the last time step is selected as the residual maneuver representation, denoted as $\mathbf{H}^r \in \mathbb{R}^{B \times d_r}$. This representation captures temporal dependencies among state-transition residuals and complements the route-level representation extracted from the raw AIS observations.

3.5. State-Adaptive Motion Aggregation and Prediction

The global navigation stream and the maneuver residual stream describe different aspects of vessel motion. Their relative importance changes with the navigation state. During steady sailing, route-level evolution usually provides the main predictive cue. During turning or speed changes, residual maneuver dynamics become more informative. A fixed fusion strategy may not capture this state-dependent relationship.

Before fusion, the two stream representations are $\mathbf{H}^g \in \mathbb{R}^{B \times d_g}$ and $\mathbf{H}^r \in \mathbb{R}^{B \times d_r}$. They are first projected into a shared fusion space:

$$\begin{aligned}\tilde{\mathbf{H}}^g &= \mathbf{H}^g \mathbf{W}_g + \mathbf{b}_g, \\ \tilde{\mathbf{H}}^r &= \mathbf{H}^r \mathbf{W}_r + \mathbf{b}_r\end{aligned}\quad (21)$$

where $\mathbf{W}_g$, $\mathbf{W}_r$, $\mathbf{b}_g$, and $\mathbf{b}_r$ are learnable parameters, and the projected representations satisfy $\tilde{\mathbf{H}}^g, \tilde{\mathbf{H}}^r \in \mathbb{R}^{B \times d_f}$, with d_f denoting the fusion dimension.

To adaptively combine the two streams, a gate vector is generated from the concatenated projected features:

$$\mathbf{g} = \sigma\left(\text{MLP}\left([\tilde{\mathbf{H}}^g; \tilde{\mathbf{H}}^r]\right)\right) \quad (22)$$

where $[\tilde{\mathbf{H}}^g; \tilde{\mathbf{H}}^r]$ denotes feature concatenation, $\text{MLP}(\cdot)$ is a multilayer perceptron, and $\mathbf{g} \in \mathbb{R}^{B \times d_f}$ is the state-adaptive gate vector. The fused representation is calculated as

$$\mathbf{H}_{fuse} = \mathbf{g} \odot \tilde{\mathbf{H}}^g + (1 - \mathbf{g}) \odot \tilde{\mathbf{H}}^r \quad (23)$$

where $\mathbf{H}_{fuse} \in \mathbb{R}^{B \times d_f}$.

The gate $\mathbf{g}$ learns a navigation-state-conditioned balance between route-level evolution and maneuver residual dynamics. This design allows the model to rely more on the global navigation stream during stable motion and to increase the contribution of the residual stream when local maneuver changes become more informative.

The fused representation $\mathbf{H}_{fuse}$ is subsequently passed to a prediction head for future trajectory generation. The prediction head uses a fully connected mapping to project the fused motion representation into the future trajectory space. Its output is the predicted latitude–longitude sequence over the next L time steps:

$$\hat{\mathbf{Y}} = F_{pred}(\mathbf{H}_{fuse}) \quad (24)$$

where $F_{pred}(\cdot)$ denotes the fully connected prediction head, and $\hat{\mathbf{Y}} \in \mathbb{R}^{B \times L \times 2}$ represents the predicted latitude–longitude sequence. Unlike recursive prediction strategies that generate future positions step by step, this design outputs the complete future sequence in one forward pass, reducing the accumulation of intermediate prediction errors.

During training, mean squared error is used to measure the difference between the predicted future trajectory and the ground-truth trajectory:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{Y}}_i - \mathbf{Y}_i\|_2^2 \quad (25)$$

where N denotes the number of training samples, $\hat{\mathbf{Y}}_i$ is the predicted future trajectory of the i -th sample, and $\mathbf{Y}_i$ is the corresponding ground-truth trajectory. Since each trajectory contains the latitude and longitude values of the next L time steps, this loss jointly penalizes prediction errors over the whole future sequence.

4. Experiments

4.1. Dataset and Preprocessing Settings

The experiments were conducted using historical AIS trajectory records collected from the Zhoushan sea area, China, a representative coastal navigation region with dense vessel traffic in eastern China. The study region covers the longitude range from 122.3° E to 123.1° E and the latitude range from 29.4° N to 30.2° N. The data span a continuous four-month period from September 2025 to December 2025. A continuous collection period was used to increase trajectory coverage and include different navigation states within the study region. Figure 6 shows the spatial distribution of AIS vessel trajectories in the Zhoushan sea area.

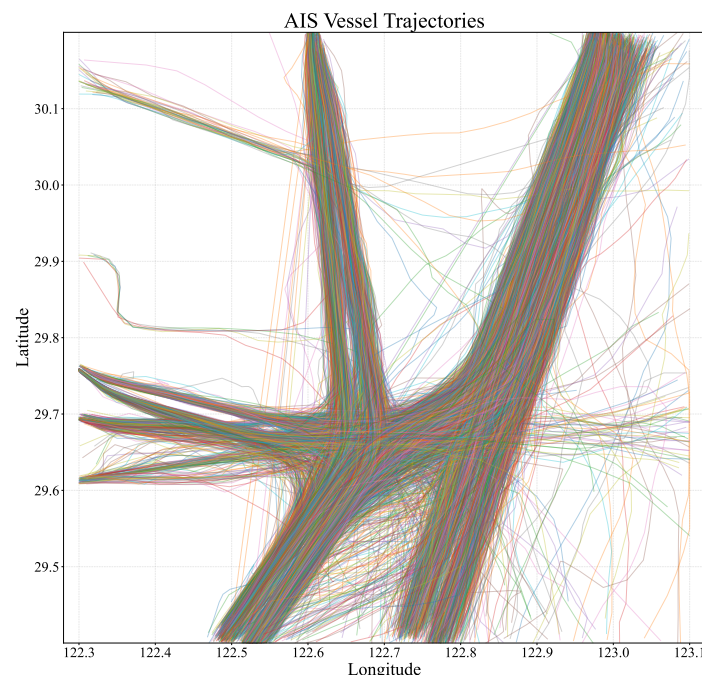


Figure 6. Spatial distribution of AIS vessel trajectories in the Zhoushan sea area.

The Zhoushan sea area was selected because it is a dense coastal navigation region with main traffic routes, port-approach waters, traffic convergence zones, and geographically constrained channels. These conditions produce both stable route-following trajectories and local maneuvering behaviors such as course adjustments, speed variations, and turns, providing a suitable real-world scenario for evaluating the proposed motion-decoupled dual-stream framework.

During preprocessing, AIS records with missing timestamps, positions, speeds, courses, or trajectory identifiers were removed. Records containing longitude or latitude values outside the study area were also excluded. The remaining records were grouped according to trajectory identifiers and arranged in chronological order to preserve the temporal order of each trajectory.

To avoid unreliable interpolation over long missing intervals, the maximum allowable time gap between two adjacent AIS records was set to 10 min. If the interval between two consecutive records exceeded this threshold, the trajectory was split at this point. Trajectory segments that could not provide sufficient observations for both the historical input window and the prediction horizon were discarded.

AIS messages are often recorded at irregular intervals. To obtain uniformly sampled historical input sequences while avoiding future information leakage, cubic spline interpolation was performed within the historical window of each prediction sample rather than on the complete trajectory segment. For a sample with prediction origin t_T , only AIS

observations not later than t_T were used to fit the spline function for the historical input window. Observations after t_T were used only to construct the supervised prediction target and were not involved in the interpolation or feature construction of the historical input.

For each AIS trajectory variable, cubic spline interpolation was applied within the historical window of the current prediction sample. In the interval $[t_i, t_{i+1}]$, the cubic spline function is defined as

$$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3, \quad t \in [t_i, t_{i+1}]. \quad (26)$$

where $S_i(t)$ denotes the interpolated value in the interval $[t_i, t_{i+1}]$, and a_i , b_i , c_i , and d_i are spline coefficients estimated from AIS observations within the historical window of the current prediction sample. The interpolated historical sequence was resampled at a fixed interval of 1 min and then used to construct the model input features. This sample-wise interpolation strategy ensures that each historical input is generated strictly from past observations, thereby avoiding information leakage from future AIS records.

After interpolation and resampling, navigation-state features were constructed from the historical AIS sequence, including latitude, longitude, speed over ground, and course over ground. The residual-stream features were constructed using position increments and Cartesian velocity components, where the latter were obtained by projecting SOG and COG into eastward and northward velocity components.

The dataset was divided into training, validation, and test sets at a ratio of 70%:15%:15% according to trajectory identifiers. All samples generated from the same trajectory were assigned to the same subset to avoid trajectory-level information leakage. Z-score normalization parameters were calculated using only the training set and then applied to the validation and test sets.

Finally, a sliding-window method was used to generate input–output samples for multi-step trajectory prediction. For each sample, the historical raw AIS sequence was used as the input of the global navigation stream, while the residual-motion sequence was used as the input of the residual maneuver stream. The prediction target was the future latitude–longitude sequence.

The complete data processing pipeline is illustrated in Figure 7. Starting from raw AIS messages, the pipeline consists of data cleaning and filtering, gap-based trajectory splitting, sample-wise historical-window interpolation and 1 min resampling, feature construction, training-set-based Z-score standardization, and sliding-window sample generation. This workflow shows how raw AIS records are transformed into the final model inputs and prediction targets.

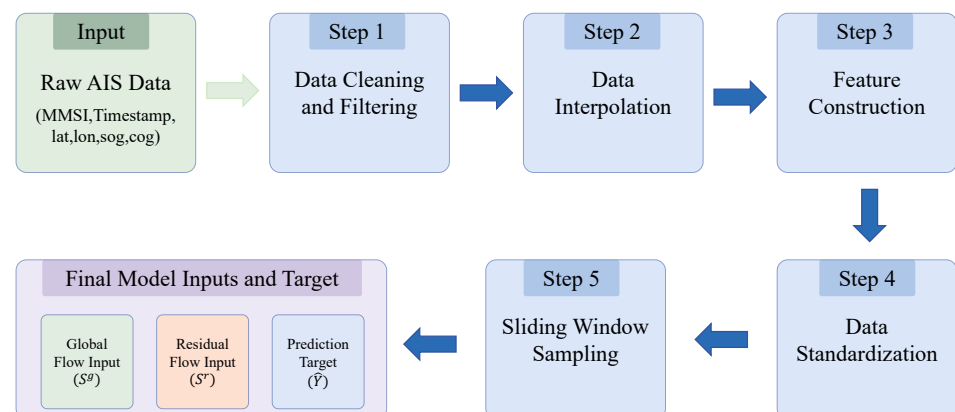


Figure 7. Workflow of AIS data preprocessing and sample construction from raw AIS messages to the final model inputs and prediction target.

4.2. Implementation Details

To improve reproducibility, the computational environment was specified as follows. All experiments were conducted on Windows 11 using Python 3.10 and an NVIDIA GeForce RTX 4060 Ti GPU (NVIDIA Corporation, Santa Clara, CA, USA). The proposed model and all baseline models were implemented in PyTorch 2.1.0 with CUDA 12.1 and cuDNN 8 acceleration. scikit-learn 1.7.2 was used for data normalization and performance evaluation, NumPy 1.25.0 and pandas 2.3.2 were used for numerical computation and AIS data preprocessing, and Matplotlib 3.10.6 was used for visualization. Unless otherwise stated, all models were trained and evaluated using the same software environment, data split, and random seed.

The proposed MD-EDTCNFormer consists of a global navigation dynamics stream and a residual maneuver dynamics stream. The EDTCN encoder in the global stream uses a hidden dimension of 64, two temporal layers, a kernel size of 3, a bottleneck ratio of 0.5, and an ECA kernel size of 3. The Transformer encoder in the residual stream uses a hidden dimension of 64, one encoder layer, and four attention heads. The gate-generation MLP takes the concatenated global-navigation and residual-maneuver features as input, with an input dimension of 128. The hidden dimension is 64 and the output gate dimension is 64. ReLU is used as the internal activation function, dropout is set to 0.1, and Sigmoid is applied to generate the final gate vector.

For training, the batch size was set to 64, and each model was trained for 120 epochs. AdamW was used as the optimizer with a learning rate of 0.001. The dropout rate was set to 0.1. The main hyperparameter settings are listed in Table 1.

Table 1. Experimental environment and parameter settings.

Category	Parameter	Setting
Global navigation stream	Encoder	EDTCN
	Hidden dimension	64
	Number of layers	2
	Kernel size	3
	Bottleneck ratio	0.5
	ECA kernel size	3
Residual maneuver stream	Encoder	Transformer
	Hidden dimension	64
	Number of layers	1
	Number of attention heads	4
Adaptive aggregation module	Input dimension	128
	Hidden dimension	64
	Output dimension	64
	Internal activation	ReLU
	Gate activation	Sigmoid
Training parameters	Batch size	64
	Epochs	120
	Optimizer	AdamW
	Learning rate	0.001
	Dropout	0.1

The baseline models were trained under the same data split, input–output setting, optimizer, learning rate, batch size, and number of epochs. For recurrent neural network baselines, the number of recurrent layers was kept consistent. For Transformer-based baselines, the number of encoder layers was also kept consistent. No additional training tricks or model-specific post-processing were used.

4.3. Evaluation Metrics

RMSE, MAE, and ADE were used to evaluate the prediction errors of different models. Since the input and output variables were normalized during training, all metrics were calculated after inverse Z-score normalization. The predicted and ground-truth trajectories were restored to the original longitude–latitude coordinate domain before error calculation.

Let N denote the total number of predicted position points in the test set. The ground-truth position of the i -th point is denoted as (x_i, y_i) , and the corresponding predicted position is denoted as $(\hat{x}_i, \hat{y}_i)$. The root mean square error is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]} \quad (27)$$

The mean absolute error is defined as

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N (|x_i - \hat{x}_i| + |y_i - \hat{y}_i|) \quad (28)$$

The average displacement error is defined as

$$\text{ADE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (29)$$

RMSE gives more weight to large coordinate deviations because of the squared error term. MAE measures the average coordinate-wise absolute deviation. ADE measures the average Euclidean distance between predicted and ground-truth positions. Lower values indicate smaller prediction errors. All reported results are expressed in the original longitude–latitude coordinate domain.

4.4. Baseline Model Results and Analysis

Six representative baseline models were selected for comparison, including LSTM, GRU, BiLSTM, LSTM-ATT, ST-Seq2Seq, and Transformer. RMSE, MAE, and ADE were used as evaluation metrics. The comparison was conducted under 5-step, 10-step, and 15-step prediction settings. Table 2 reports the prediction errors of all models on the test set. The corresponding trajectory visualizations are provided in Figure 8.

LSTM [29]: Uses gated recurrent units to learn temporal dependencies from historical AIS sequences and maps the hidden representation to future positions through a fully connected layer.

GRU [48]: Adopts GRU for trajectory sequence learning. Compared with LSTM, GRU has a simpler structure and fewer parameters, enabling lower computational complexity.

BiLSTM [30]: Employs a bidirectional LSTM to extract temporal features from historical trajectories in both forward and backward directions, then fuses bidirectional hidden states for prediction, enhancing contextual representation.

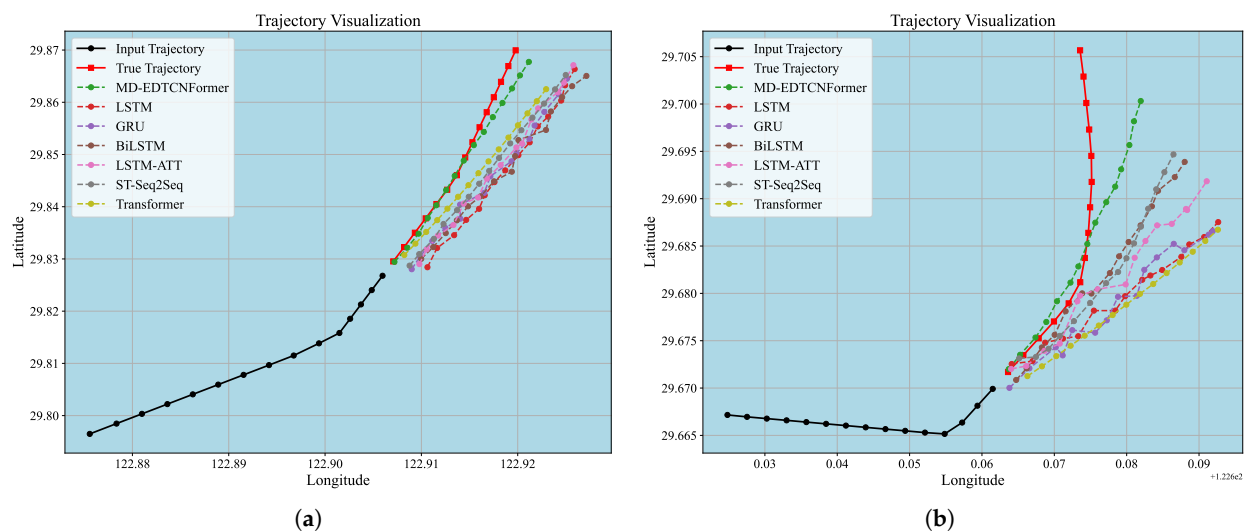
LSTM-ATT [49]: Introduces an attention mechanism into LSTM, assigning different weights to hidden states at different historical steps to focus on key moments relevant to future trajectory changes.

ST-Seq2Seq [31]: A spatiotemporal feature-optimized Seq2Seq framework. It uses a GRU-based encoder–decoder to integrate spatiotemporal correlations: the encoder represents historical trajectories, and the decoder generates future multi-step predictions.

Transformer [50]: Uses self-attention to capture global dependencies in input trajectories. Multi-head attention, residual connections, layer normalization, and feed-forward networks jointly extract long-range temporal relationships for future prediction.

Table 2. Comparison of different baseline models on the test set.

Prediction Step	Methods	RMSE	MAE	ADE
5	LSTM	0.00180	0.00095	0.00149
	GRU	0.00160	0.00092	0.00146
	BiLSTM	0.00149	0.00078	0.00122
	LSTM-ATT	0.00164	0.00091	0.00142
	ST-Seq2Seq	0.00122	0.00068	0.00106
	Transformer	0.00159	0.00113	0.00179
	MD-EDTCNFormer	0.00054	0.00028	0.00045
10	LSTM	0.00239	0.00125	0.00197
	GRU	0.00214	0.00116	0.00182
	BiLSTM	0.00204	0.00109	0.00171
	LSTM-ATT	0.00251	0.00135	0.00212
	ST-Seq2Seq	0.00173	0.00094	0.00148
	Transformer	0.00194	0.00125	0.00198
	MD-EDTCNFormer	0.00122	0.00061	0.00093
15	LSTM	0.00319	0.00168	0.00263
	GRU	0.00293	0.00161	0.00253
	BiLSTM	0.00260	0.00135	0.00212
	LSTM-ATT	0.00291	0.00162	0.00251
	ST-Seq2Seq	0.00244	0.00129	0.00202
	Transformer	0.00289	0.00178	0.00259
	MD-EDTCNFormer	0.00190	0.00094	0.00148

**Figure 8.** Comparison of predicted trajectories of different baseline models in straight-route and curved-route scenarios. (a) Straight-route case; (b) curved-route case.

The comparison results demonstrate that MD-EDTCNFormer achieves the best prediction performance under all forecasting horizons. Among the baseline models, ST-Seq2Seq generally achieves the best performance, while recurrent-based models such as BiLSTM, GRU, LSTM-ATT, and LSTM show competitive prediction capability. Transformer performs well in capturing long-range temporal dependencies but exhibits relatively higher errors in short-to-medium-horizon prediction. This indicates that different temporal modeling mechanisms have distinct advantages for vessel trajectory prediction under different forecasting conditions.

Compared with conventional recurrent models, ST-Seq2Seq benefits from its explicit encoder–decoder structure. The encoder summarizes historical trajectory information,

while the decoder generates future trajectory sequences based on the learned temporal representation. In contrast, LSTM and GRU mainly rely on recurrent hidden states to accumulate temporal information, and their sequential propagation may gradually weaken the influence of earlier motion states as the prediction horizon increases. BiLSTM improves contextual representation by incorporating information from both temporal directions, while LSTM-ATT enhances feature selection through attention mechanisms. However, these recurrent models still encode heterogeneous motion patterns within a unified temporal representation space and do not explicitly distinguish between stable navigation evolution and local maneuver variations.

The standard Transformer provides strong capability for modeling global temporal dependencies through self-attention, but it shows relatively weaker performance than ST-Seq2Seq in short-to-medium-horizon prediction. Specifically, in the five-step prediction scenario, Transformer exhibits higher RMSE, MAE, and ADE values than ST-Seq2Seq, with increases of approximately 30.33%, 66.18%, and 68.87%, respectively. This difference suggests that standalone self-attention may not be optimal for capturing immediate local vessel dynamics. Although self-attention enables interactions among all historical time steps, it does not explicitly preserve the strong local continuity inherent in vessel motion. For short-term prediction, recent observations usually contain dominant information related to current velocity, heading variation, and maneuver tendency. However, the global attention mechanism may distribute attention among distant historical states that have weaker relevance to near-future motion, thereby reducing sensitivity to local dynamic changes. In comparison, the hidden-state transition mechanism of encoder–decoder recurrent structures naturally maintains sequential continuity and is more suitable for modeling short-term trajectory evolution.

As the prediction horizon increases, accumulated uncertainty becomes a critical challenge for all forecasting models. Compared with the strongest baseline ST-Seq2Seq, MD-EDTCNFormer reduces RMSE, MAE, and ADE by approximately 29.48%, 35.11%, and 37.16% in the 10-step prediction task. For the longer 15-step prediction task, MD-EDTCNFormer still achieves reductions of 22.13%, 27.13%, and 26.73% compared with ST-Seq2Seq, although the improvement margin decreases due to the increasing uncertainty of future trajectories. These results indicate that the proposed framework can effectively alleviate error accumulation by separating different motion components and learning their respective temporal representations.

The performance improvement of MD-EDTCNFormer mainly originates from its motion-decoupled representation strategy. Instead of forcing heterogeneous vessel dynamics into a single latent space, the proposed framework separately models route-level evolution and maneuver-related residual variations. The global navigation stream focuses on stable sailing trends, while the residual maneuver stream captures short-term state transitions caused by turning, acceleration, and speed changes. Through state-adaptive motion aggregation, the model dynamically adjusts the contribution of different motion components according to navigation conditions.

Figure 8 compares the predicted trajectories of different models in straight-route and curved-route cases. In the straight-route case, most models follow the general sailing direction, but endpoint deviations are still visible. In the curved-route case, the differences among models become clearer because the prediction requires a better description of trajectory curvature and turning tendency. MD-EDTCNFormer produces trajectories closer to the ground truth in both cases, especially in the curved-route scenario. This visual comparison is consistent with the numerical results in Table 2 and supports the role of residual maneuver modeling in trajectory prediction.

4.5. Ablation Study

An ablation study was conducted to examine the contribution of the main components in MD-EDTCNFormer. Four models were compared: TCN, EDTCN, TCNFormer, and MD-EDTCNFormer. TCN uses a standard temporal convolutional encoder as the baseline. EDTCN replaces the standard temporal convolution with the proposed lightweight temporal encoder, where depthwise separable causal convolution and ECA are jointly used in the Light Temporal Block. TCNFormer keeps the basic TCN encoder and adds a Transformer-based residual stream. MD-EDTCNFormer combines the EDTCN encoder, the residual maneuver stream, and state-adaptive motion aggregation.

Since depthwise separable causal convolution and ECA are jointly designed within the Light Temporal Block, EDTCN is treated as an integrated temporal encoder rather than separating these two operations into independent plug-in modules. The ablation results are reported in Table 3.

Table 3. Ablation results of different model components.

Model	RMSE	MAE	ADE	FLOPs	Params
TCN	0.00236	0.00120	0.00189	1,394,910	40,158
EDTCN	0.00207	0.00101	0.00160	344,862	13,476
TCNFormer	0.00204	0.00102	0.00158	2,977,310	107,998
MD-EDTCNFormer	0.00190	0.00094	0.00148	1,927,262	81,316

Compared with TCN, EDTCN reduces RMSE, MAE, and ADE by 12.29%, 15.83%, and 15.34%, respectively. Meanwhile, the parameter number decreases from 40,158 to 13,476 and the FLOPs are reduced by 75.28%. This improvement is mainly attributed to the lightweight design of the Light Temporal Block. Standard temporal convolution jointly models temporal dependencies and channel interactions, resulting in redundant parameter connections and computational cost. In contrast, EDTCN adopts depthwise separable causal convolution to decouple temporal feature extraction and channel fusion. The depthwise convolution captures temporal patterns within each channel, while the pointwise convolution performs efficient cross-channel fusion, reducing both parameter redundancy and FLOPs while preserving multi-scale temporal modeling capability.

In addition, the ECA module improves feature representation by adaptively recalibrating channel responses. Different AIS variables contribute differently under various navigation conditions, and ECA can emphasize informative motion features while suppressing less relevant responses with limited overhead. Combined with bottleneck feature compression, EDTCN achieves improved route-level representation with fewer parameters and lower computational complexity.

The comparison between TCN and TCNFormer demonstrates the contribution of the residual maneuver stream. TCNFormer reduces RMSE, MAE, and ADE by 13.56%, 15.00%, and 16.40%, respectively, indicating that state-transition residuals provide complementary motion information beyond the original navigation-state sequence. However, introducing the Transformer residual branch increases the parameter number from 40,158 to 107,998 and also leads to a clear increase in FLOPs. This increase mainly results from the additional Transformer encoder, including the self-attention projection layers and feed-forward network. Since TCNFormer retains the original TCN encoder while adding an independent residual encoder, the parameters and computations of the two branches are accumulated.

Among all variants, MD-EDTCNFormer achieves the best prediction performance. Compared with TCN, it reduces RMSE, MAE, and ADE by 19.49%, 21.67%, and 21.69%, respectively. Compared with EDTCN, further reductions of 8.21%, 6.93%, and 7.50% are obtained. These improvements demonstrate that residual maneuver information effec-

tively complements the route-level features extracted by EDTCN. The global navigation branch captures stable motion evolution, whereas the residual branch focuses on local maneuver variations.

Compared with TCNFormer, MD-EDTCNFormer reduces RMSE, MAE, and ADE by 6.86%, 7.84%, and 6.33%, respectively. Meanwhile, the parameter number decreases from 107,998 to 81,316 and the FLOPs are reduced by 35.27%. This reduction is achieved through efficient architectural design. Specifically, EDTCN replaces the original TCN encoder with a lightweight temporal encoder, reducing redundant parameters and computations through depthwise separable convolution and bottleneck compression. Meanwhile, the compact Transformer residual branch maintains the ability to model maneuver dynamics without excessive complexity, and the state-adaptive aggregation module fuses two-stream features in a shared feature space to avoid unnecessary parameter and FLOP growth.

The visual results in Figure 9 are consistent with the quantitative findings. As shown in Figure 9a, MD-EDTCNFormer produces trajectories that most closely follow the ground-truth paths, whereas TCN and EDTCN exhibit relatively larger prediction deviations. Figure 9b compares the model complexity of the ablation variants. EDTCN has the lowest parameter count and computational cost, while TCNFormer introduces the highest complexity because of its additional Transformer-based residual branch. In contrast, MD-EDTCNFormer achieves the best prediction accuracy while maintaining a relatively compact and computationally efficient architecture.

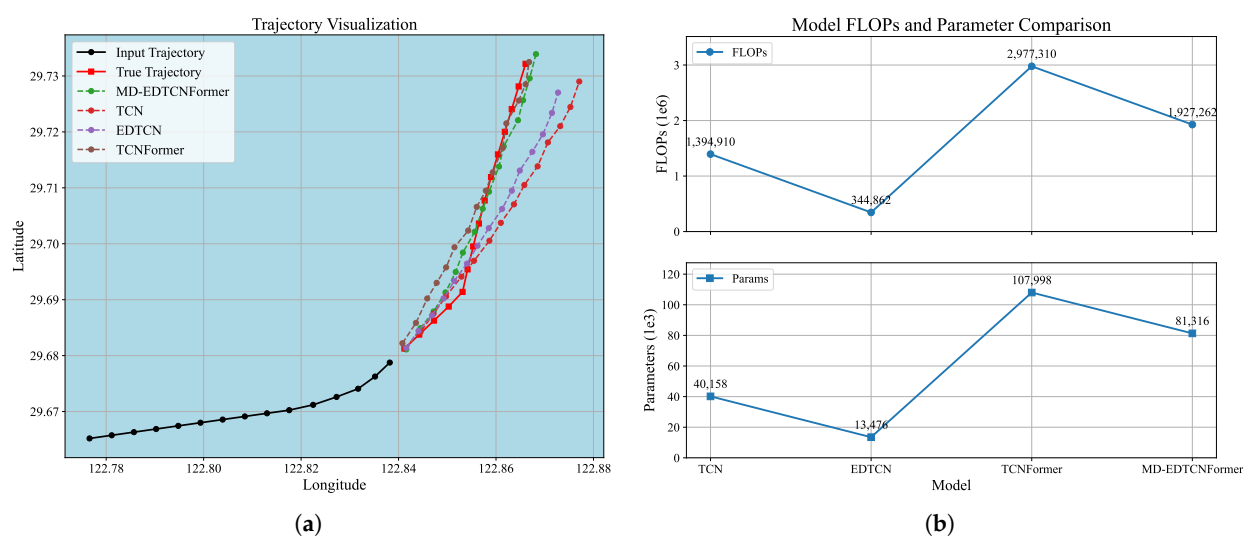


Figure 9. Comparison of predicted trajectories and parameter sizes of ablation models. of predicted trajectories and parameter sizes of ablation models. (a) Trajectory prediction results of different ablation models; (b) comparison of Params and FLOPs.

4.6. Effect of Residual-Stream Encoder Selection

To examine the influence of the residual maneuver stream, several model variants were constructed by keeping the global navigation stream unchanged and replacing only the encoder used in the residual stream. GRU, LSTM, EDTCN, and Transformer were used to encode the residual motion sequence, forming EDTCN-GRU, EDTCN-LSTM, EDTCN-EDTCN, and MD-EDTCNFormer, respectively. The results are reported in Table 4.

Table 4. Prediction performance with different residual-stream encoders.

Model	RMSE	MAE	ADE
EDTCN-GRU	0.00195	0.00097	0.00153
EDTCN-LSTM	0.00200	0.00100	0.00158
EDTCN-EDTCN	0.00197	0.00098	0.00155
MD-EDTCNFormer	0.00190	0.00094	0.00148

The results in Table 4 indicate that the choice of residual-stream encoder has a direct influence on prediction accuracy. Among the non-Transformer variants, EDTCN-GRU achieves the best overall performance, suggesting that the compact recurrent structure of GRU is effective for modeling short-term dependencies in residual motion sequences. EDTCN-EDTCN obtains comparable results, which indicates that convolutional encoding can also extract local residual patterns from state-transition features. In contrast, EDTCN-LSTM shows slightly higher errors, implying that the more complex gating structure of LSTM does not provide a clear advantage for the constructed residual sequence under the current experimental setting.

MD-EDTCNFormer achieves the lowest prediction errors among all residual-stream encoder variants. Compared with EDTCN-GRU, the strongest non-Transformer variant, MD-EDTCNFormer reduces RMSE, MAE, and ADE by 2.56%, 3.09%, and 3.27%, respectively. Although the improvement margin is moderate, the reduction is consistent across all evaluation metrics, indicating that the Transformer encoder provides a more suitable mechanism for residual maneuver modeling.

The advantage of the Transformer-based residual stream is mainly related to the characteristics of maneuver residuals. Unlike route-level navigation trends, residual motion features are more sensitive to abrupt and non-stationary changes, such as turning, acceleration, and velocity-direction adjustment. These changes may not be determined only by the immediately preceding state, but can also be associated with several earlier residual variations within the observation window. Recurrent encoders such as GRU and LSTM process residual sequences step by step, which is effective for preserving local continuity but may limit direct interaction between non-adjacent maneuver states. Convolutional encoders rely on fixed receptive fields, making them suitable for local pattern extraction but less flexible in capturing irregular temporal dependencies. In contrast, self-attention allows each residual state to directly attend to other historical residual states, enabling the model to identify important maneuver-related transitions without assuming fixed temporal distances.

Figure 10 compares the training and validation loss curves of the four residual-stream encoder variants under the same optimization setting. All models exhibit a rapid loss reduction during the early training stage, followed by a gradual convergence to a stable level, indicating that the different encoder structures can all effectively learn from the residual-motion sequence. Although small fluctuations appear in the intermediate stage, the training and validation curves remain close after convergence and show no persistent divergence, suggesting that none of the variants suffers from obvious overfitting or unstable optimization. Among the four models, MD-EDTCNFormer converges to the lowest validation-loss level, which is consistent with its superior RMSE, MAE, and ADE results in Table 4. This agreement indicates that the performance improvement mainly originates from the stronger temporal representation capability of the Transformer-based residual encoder rather than differences in convergence behavior.

Overall, GRU, LSTM, and EDTCN can all encode residual motion information to some extent, but the Transformer encoder provides more flexible temporal dependency modeling for non-stationary maneuver residuals. Therefore, the Transformer-based residual stream

is adopted in MD-EDTCNFormer to complement the route-level representation learned by the global navigation stream.

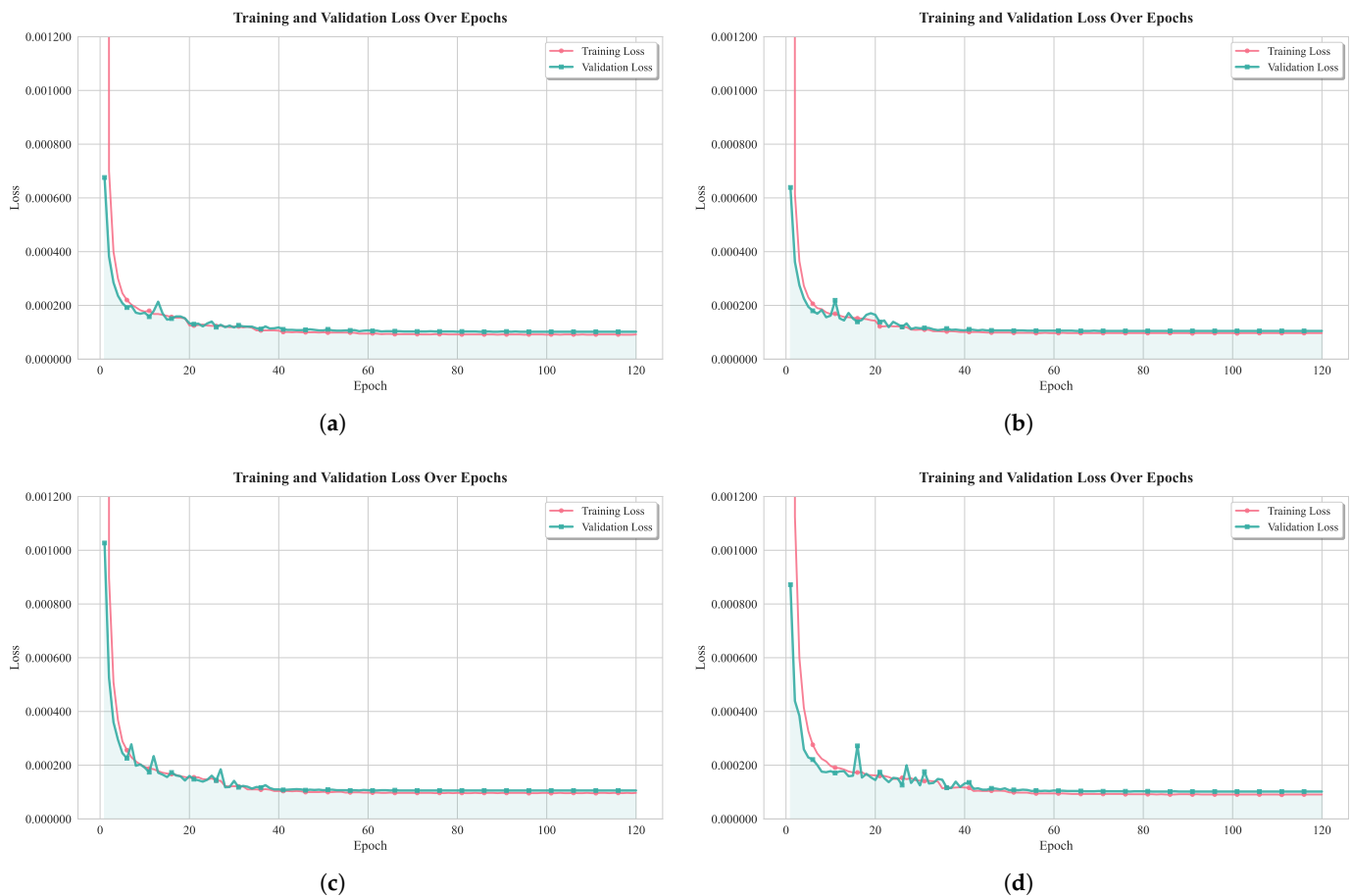


Figure 10. Comparison of training and validation loss curves of MD-EDTCNFormer and its ablation variants in the training process. (a) EDTCN-GRU case; (b) EDTCN-LSTM case; (c) EDTCN-EDTCN case; (d) MD-EDTCNFormer case.

5. Conclusions

This study presented MD-EDTCNFormer, a motion-decoupled dual-stream framework for AIS-based vessel trajectory prediction. The model separates vessel motion representation into two complementary components: route-level navigation evolution and maneuver-related residual dynamics. The global navigation stream learns temporal evolution from raw AIS states, while the residual maneuver stream models local state-transition residuals. A state-adaptive aggregation module is then used to combine the two representations for multi-step trajectory prediction.

Experimental results on the Zhoushan coastal dataset are reported in terms of RMSE, MAE, and ADE. MD-EDTCNFormer yields lower error values than the compared baselines across all prediction horizons. For the 15-step prediction task, the RMSE and ADE of MD-EDTCNFormer are 22.13% and 26.73% lower than those of ST-Seq2Seq. Visualization results show that the predicted trajectories of MD-EDTCNFormer are closer to the ground truth than those of other models in curved and turning scenarios.

The proposed framework may be relevant to maritime traffic supervision applications. The dual-stream design and state-adaptive gate offer structural features that could support maneuvering-risk identification in VTS contexts. The model's modest parameter size and

computational requirements are also favorable for edge-side processing, which may be beneficial for real-time AIS-based monitoring.

Despite these contributions, several limitations remain to be addressed. First, the current model relies solely on AIS kinematic variables and does not incorporate external environmental factors or vessel-specific attributes. Second, the model was trained and validated only on the Zhoushan sea area; its transferability to regions with distinct traffic density and route structures requires further cross-region testing. Possible extensions include introducing a multi-source environmental fusion branch as a third stream to the current adaptive aggregation module, and incorporating an interaction-aware module to capture multi-vessel dependencies in congested waterways.

Author Contributions: Methodology, C.W. and S.Z.; Software, D.Z. and Z.L.; Validation, Y.Z., R.W., B.W. and Q.N.; Data curation, R.W. and S.Z.; Writing—original draft, Y.Z.; Writing—review and editing, C.W., S.Z. and Q.N.; Visualization, R.W.; Supervision, Q.N.; Project administration, S.Z. and Z.L.; Funding acquisition, Q.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fujian University–Industry Cooperation Project under the projects ‘Green and Intelligent Ship in the Fujian Region’ (grant number CBG4N21-4-4) and ‘Development and Application of a Multi-Source-Data Intelligent Ship Auxiliary Terminal Based on Large-Model Technology’ (grant number 2024H6020).

Data Availability Statement: The datasets that were used and analyzed in this study can be obtained from the corresponding author upon reasonable request.

Conflicts of Interest: Zhenjun Li was employed by Fujian Province Shipbuilding Industry Group Co., Ltd. Bingkun Wu was employed by Zhongshu (Xiamen) Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Yan, R.; Wang, S.; Zhen, L.; Laporte, G. Emerging approaches applied to maritime transport research: Past and future. *Commun. Transp. Res.* **2021**, *1*, 100011.
2. Li, Y.; Yu, Q.; Yang, Z. Vessel trajectory prediction for enhanced maritime navigation safety: A novel hybrid methodology. *J. Mar. Sci. Eng.* **2024**, *12*, 1351.
3. Gan, L.; Ye, B.; Huang, Z.; Xu, Y.; Chen, Q.; Shu, Y. Knowledge graph construction based on ship collision accident reports to improve maritime traffic safety. *Ocean Coast. Manag.* **2023**, *240*, 106660.
4. Cai, J.; Chen, G.; Lützen, M.; Rytter, N.G.M. A practical AIS-based route library for voyage planning at the pre-fixtured stage. *Ocean Eng.* **2021**, *236*, 109478.
5. Li, H.; Jiao, H.; Yang, Z. AIS data-driven ship trajectory prediction modelling and analysis based on machine learning and deep learning methods. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *175*, 103152.
6. Yang, Y.; Liu, Y.; Li, G.; Zhang, Z.; Liu, Y. Harnessing the power of machine learning for AIS data-driven maritime research: A comprehensive review. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *183*, 103426.
7. Perera, L.P.; Soares, C.G. Ocean vessel trajectory estimation and prediction based on extended Kalman filter. In *Proceedings of the Second International Conference on Adaptive and Self-Adaptive Systems and Applications*; Citeseer: University Park, PA, USA, 2010; pp. 14–20.
8. Perera, L.P.; Oliveira, P.; Soares, C.G. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1188–1200.
9. Rong, H.; Teixeira, A.P.; Soares, C.G. Ship trajectory uncertainty prediction based on a Gaussian process model. *Ocean Eng.* **2019**, *182*, 499–511.
10. Capobianco, S.; Millefiori, L.M.; Forti, N.; Braca, P.; Willett, P. Deep learning methods for vessel trajectory prediction based on recurrent neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 4329–4346.
11. Bao, K.; Bi, J.; Gao, M.; Sun, Y.; Zhang, X.; Zhang, W. An improved ship trajectory prediction based on AIS data using MHA-BiGRU. *J. Mar. Sci. Eng.* **2022**, *10*, 804.

12. Zhang, X.; Liu, J.; Gong, P.; Chen, C.; Han, B.; Wu, Z. Trajectory prediction of seagoing ships in dynamic traffic scenes via a gated spatio-temporal graph aggregation network. *Ocean Eng.* **2023**, *287*, 115886.
13. Xue, H.; Wang, S.; Xia, M.; Guo, S. G-Trans: A hierarchical approach to vessel trajectory prediction with GRU-based transformer. *Ocean Eng.* **2024**, *300*, 117431.
14. Qiang, H.; Guo, Z.; Xie, S.; Peng, X. Mstformer: Motion inspired spatial-temporal transformer with dynamic-aware attention for long-term vessel trajectory prediction. *arXiv* **2023**, arXiv:2303.11540.
15. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
16. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Hartwig, A. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
17. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: New York, NY, USA, 2020; pp. 11534–11542.
18. Zhou, T.; Jiang, D.; Lin, Z.; Han, G.; Xu, X.; Qin, J. Hybrid dual Kalman filtering model for short-term traffic flow forecasting. *IET Intell. Transp. Syst.* **2019**, *13*, 1023–1032.
19. Jaskólski, K. Automatic identification system (AIS) dynamic data estimation based on discrete Kalman filter (KF) algorithm. *Zesz. Nauk. Akad. Mar. Wojennej* **2017**, *58*, 71–87.
20. Guo, S.; Liu, C.; Guo, Z.; Feng, Y.; Hong, F.; Huang, H. Trajectory prediction for ocean vessels base on K-order multivariate Markov chain. In *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*; Springer: Cham, Switzerland, 2018; pp. 140–150.
21. Liu, C.; Guo, S.; Feng, Y.; Hong, F.; Huang, H.; Guo, Z. L-VTP: Long-term vessel trajectory prediction based on multi-source data analysis. *Sensors* **2019**, *19*, 4365.
22. Nie, Z.; Shen, F.; Xu, D.; Li, Q. An EMD-SVR model for short-term prediction of ship motion using mirror symmetry and SVR algorithms to eliminate EMD boundary effect. *Ocean Eng.* **2020**, *217*, 107927.
23. Borkowski, P. The ship movement trajectory prediction algorithm using navigational data fusion. *Sensors* **2017**, *17*, 1432.
24. Tu, E.; Zhang, G.; Mao, S.; Rachmawati, L.; Huang, G.-B. Modeling historical AIS data for vessel path prediction: A comprehensive treatment. *arXiv* **2020**, arXiv:2001.01592.
25. Zhang, C.; Bin, J.; Wang, W.; Peng, X.; Wang, R.; Haldearn, R.; Liu, Z. AIS data driven general vessel destination prediction: A random forest based approach. *Transp. Res. Part C Emerg. Technol.* **2020**, *118*, 102729.
26. Ma, L.; Guo, Z.; Shi, G. AIS data driven ship behavior modeling in fairways: A random forest based approach. *Appl. Sci.* **2024**, *14*, 8484.
27. Liu, R.W.; Liang, M.; Nie, J.; Lim, W.Y.B.; Zhang, Y.; Guizani, M. Deep learning-powered vessel trajectory prediction for improving smart traffic services in marine Internet of Things. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 3080–3094.
28. Slaughter, I.; Charla, J.L.; Siderius, M.; Lipor, J. Vessel trajectory prediction with recurrent neural networks: An evaluation of datasets, features, and architectures. *J. Ocean Eng. Sci.* **2025**, *10*, 229–238.
29. Tang, H.; Yin, Y.; Shen, H. A model for vessel trajectory prediction based on long short-term memory neural network. *J. Mar. Eng. Technol.* **2022**, *21*, 136–145.
30. Yang, C.H.; Wu, C.H.; Shao, J.C.; Wang, Y.C.; Hsieh, C.M. AIS-based intelligent vessel trajectory prediction using Bi-LSTM. *IEEE Access* **2022**, *10*, 24302–24315.
31. You, L.; Xiao, S.; Peng, Q.; Claramunt, C.; Han, X.; Guan, Z.; Zhang, J. ST-Seq2Seq: A spatio-temporal feature-optimized Seq2Seq model for short-term vessel trajectory prediction. *IEEE Access* **2020**, *8*, 218565–218574.
32. Jiang, Y.; Chang, Y.; Zhang, B.; Ge, Y.; Zhao, X. A review of vessel trajectory prediction research: Methodological evolution, current challenges, and future directions. *Int. J. Nav. Arch. Ocean Eng.* **2026**, *18*, 100754.
33. Liu, X.; Yuan, H.; Xiao, C.; Wang, Y.; Yu, Q. Hybrid-driven vessel trajectory prediction based on uncertainty fusion. *Ocean Eng.* **2022**, *248*, 110836.
34. Liu, R.W.; Hu, K.; Liang, M.; Li, Y.; Liu, X.; Yang, D. QSD-LSTM: Vessel trajectory prediction using long short-term memory with quaternion ship domain. *Appl. Ocean Res.* **2023**, *136*, 103592.
35. Wu, W.; Chen, P.; Chen, L.; Mou, J. Ship trajectory prediction: An integrated approach using ConvLSTM-based sequence-to-sequence model. *J. Mar. Sci. Eng.* **2023**, *11*, 1484.
36. Lin, Z.; Yue, W.; Huang, J.; Wan, J. Ship trajectory prediction based on the TTCN-attention-GRU model. *Electronics* **2023**, *12*, 2556.
37. Gan, L.; Gao, Z.; Zhang, X.; Xu, W.; Cai, Y.; Song, L. Maritime vessel trajectory prediction based on sequential long short-term memory and U-Net architectures. *Ocean Eng.* **2025**, *334*, 121598.
38. Yin, J.; Yu, Z.; Wu, H. Ship trajectory prediction based on LSTM model with multi-scale convolution and attention mechanism. *Ocean Eng.* **2025**, *338*, 122055.
39. Jiang, J.; Zuo, Y. STIA-DJANet: Spatial-temporal intention-aware vessel trajectory prediction based on dual-joint attention network for e-navigation. *Expert Syst. Appl.* **2025**, *262*, 125550.

40. Xiong, C.; Li, J.; Wu, X.; Liu, D.; Wang, Q.; Gao, R.; Ruan, J. Enhancing vessel trajectory prediction with bi-state space and squeeze excitation attention. *Ocean Eng.* **2025**, *336*, 121688.
41. Jiang, D.; Shi, G.; Li, N.; Ma, L.; Li, W.; Shi, J. TRFM-LS: Transformer-based deep learning method for vessel trajectory prediction. *J. Mar. Sci. Eng.* **2023**, *11*, 880.
42. Qiang, H.; Guo, Z.; Chu, Z.; Xie, S.; Peng, X. Motion-inspired spatial-temporal Transformer for accurate vessel trajectory prediction. *Eng. Appl. Artif. Intell.* **2025**, *148*, 110391.
43. Cao, C.; Hu, H.X. SGAformer: A granularity-aware Transformer with skip-pyramid attention mechanism for port vessel trajectory prediction. *Expert Syst. Appl.* **2026**, 131281.
44. Zhao, J.; Yan, Z.; Zhou, Z.; Chen, X.; Wu, B.; Wang, S. A ship trajectory prediction method based on GAT and LSTM. *Ocean Eng.* **2023**, *289*, 116159.
45. Zhang, R.; Chen, X.; Ye, L.; Yu, W.; Zhang, B.; Liu, J. Predicting vessel trajectories using ASTGCN with StemGNN-derived correlation matrix. *Appl. Sci.* **2024**, *14*, 4104.
46. Wu, Y.; Yv, W.; Zeng, G.; Shang, Y.; Liao, W. GL-STGCNN: Enhancing multi-ship trajectory prediction with MPC correction. *J. Mar. Sci. Eng.* **2024**, *12*, 882.
47. Gong, J.; Li, H.; Jiao, H.; Yang, Z. Uncertainty-aware ship trajectory prediction via spatio-temporal graph transformer. *Transp. Res. Part E* **2025**, *203*, 104315.
48. Cho, K.; Van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1724–1734.
49. Zhang, J.; Liu, J.; Lin, T.; Dai, W. A sequence-to-sequence based multi-semantic network with attention for long-term vessel trajectory prediction. In *Proceedings of the 2024 2nd International Conference on Computer, Vision and Intelligent Technology*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–5.
50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.

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