

Multiple Variational Kalman-GRU for Ship Trajectory Prediction with Uncertainty

Chengfeng Jia, Jie Ma, Wouter M. Kouw

Abstract—Accurate prediction of ship trajectories is crucial for ensuring safe and efficient navigation. However, predicting ship trajectories in complex and dynamic environments presents significant challenges. Ships exhibit multi-mode motions, manifesting as diverse motion patterns even under similar circumstances, influenced by factors such as navigational intentions and operational tasks. Moreover, trajectory prediction is further complicated by time-varying ship dynamics, encompassing sailing conditions, ship maneuvering, and environmental factors. In this paper, we propose a Bayesian multiple model with an online model selection strategy to dynamically represent the latent motion mode from early observations. Each sub-model integrates a variational Kalman filter and Gated Recurrent Unit (GRU) neural network, enabling the estimation of time-varying transition coefficients and the process noise specific to different motion modalities. This hybrid methodology leverages the strengths of probabilistic recursive estimation of the Kalman filter while benefiting from the capacity of a GRU network to learn complex temporal dependencies from historical data. The proposed method was evaluated on ship trajectories across different observation lengths and prediction horizons and outperformed the baseline in terms of both accuracy and plausibility.

Index Terms—ship trajectory prediction, ship navigation, Kalman filter, deep learning, variational inference.

I. INTRODUCTION

The shipping sector has been the world’s primary facilitator of global trade, with waterway transportation playing a significant role, encompassing over 90% of total transport. [1]. The field of waterway transportation is undergoing a technological revolution, referred to as Shipping 4.0 [2], necessitating the adoption and integration of modern technologies, including intelligent computational and advanced automation technology, into existing navigation systems to enhance safety and efficiency. An essential aspect of Shipping 4.0 is to improve situational awareness, which will enhance the safety of maritime operations. Situational awareness for ship navigation

Chengfeng Jia is affiliated with School of Navigation, Wuhan University of Technology, Wuhan 430063, China, Centre for Advanced Robotics Technology Innovation, NTU, Singapore 639798, and Department of Electrical Engineering, TU Eindhoven, Eindhoven 5600MB, the Netherlands (email: c.jia@tue.nl).

Jie Ma is affiliated with State Key Laboratory of Maritime Technology and Safety, and School of Navigation, Wuhan University of Technology, Wuhan 430063, China (email: majie@whut.edu.cn)

Wouter M. Kouw is affiliated with Department of Electrical Engineering, TU Eindhoven, Eindhoven 5600MB, the Netherlands (email: w.m.kouw@tue.nl).

This work was supported in part by NSF, China, under Grant 52271366 and Grant 51679182; and in part by Jiangsu Province Science and Technology Achievement Transformation Special Fund Project(BA2022014); and in part by National Research Foundation of Singapore under its Medium-Sized Center for Advanced Robotics Technology Innovation and by Naval Group Far East Pte Ltd via an RCA with NTU. (Corresponding author: Jie Ma)

was defined as understanding what is happening now and what will happen in the near-future [3]. In this context, numerous recent studies have developed precise models to predict a ship’s future trajectory, which supports early collision warning systems, abnormal behavior detection, and collision avoidance decision-making.

Ship trajectory prediction methods can generally be categorized into two groups: model-based and data-driven approaches. The model-based approaches aim to predict future positions based on dynamical systems that depend on laws of physics. The Kalman filter is a preferred technique because it provides a recursive way for ship motion state estimation and quantifies uncertainties [4]. However, the Kalman filter requires model specifications, such as the transition matrix, process noise, and measurement noise, which need to be calibrated by the user. Ship motion is a time-varying dynamic system that depends on sailing conditions, control inputs, as well as environmental factors [5], [6]. Simplification of model specifications, such as assuming the transition matrix is time-invariant, may lead to insufficient performance.

On the other hand, data-driven methods exploit the benefits of large-scale datasets to achieve higher prediction accuracy without relying on expert knowledge. Deep learning techniques, in particular, are increasingly applied to construct trajectory prediction models by implicitly determining the relationship between the observations and predictions. However, this prediction tends to return the average output when given similar observations [7]. Due to uncertain future destinations, ships could perform differently under the same circumstances. The performance of a single model may quickly deteriorate when the ship trajectories exhibit multi-modality, as is often the case in intersection waterways [8]. The ships follow similar trajectories before they sail into the intersection but their trajectories diverge when they sail into the target branch or destinations [9]. In such scenarios, averages may not be optimal solutions. Accurately predicting the trajectories of target ships as they navigate towards specific destinations is crucial, especially in collision avoidance applications. This enables own ship to avoid potential conflicts by steering clear of the paths of target vessels and taking necessary evasive actions at the intersection.

To address the challenges posed by the time-varying dynamics and multi-mode nature of ship motions, we propose a Bayesian multiple model (MK-GRU) with Kalman-GRU (K-GRU) submodels. Each sub-model captures the dynamics of a specific motion mode. The sub-models bring together the best of both worlds by augmenting the Kalman filter with Gated Recurrent Unit (GRU) neural network. GRU, a

type of recurrent neural network, effectively handles temporal dependencies in trajectory with fewer parameters, ensuring computational efficiency. Consequently, GRU is utilized to track the evolution of transition coefficients in the Kalman filter over time.

In summary, our contributions are:

- We present a Bayesian multiple model approach to address the multi-mode nature of ship motions, incorporating an online model selection strategy for dynamic representation of the hidden motion modes.
- A hybrid model, the Kalman-GRU, is proposed for sub-models, with a focus on simultaneously estimating the time-varying transition coefficients and process noise of one specific mode of ship dynamics by variational inference.
- We investigate the effect of ship trajectory observation length on accuracy for various prediction horizon lengths. The results demonstrate that the proposed model surpasses the performance of existing deep learning models. Moreover, we conducted a comparative analysis between single-model and multi-model. The results revealed that employing a multi-model approach with a reasonable number of models can reduce prediction errors.

II. RELATED WORK

An example of the effectiveness of physics-based model-driven approaches is the trajectory prediction model in Wiig et al. that considers parameters such as mass, force, damping, and disturbance [10]. Similarly, Zhang et al. [11] proposed a maneuvering model, which introduced a velocity potential to decompose the maneuvering motion and the wave-induced motion. Perera et al. pointed out that one single model may not be enough to capture the complexity of ship dynamics, and developed a multiple model based on extended Kalman filters corresponding to specific modes of operation [12]. Still, there remain various sources of uncertainty in real navigation scenarios, such as environmental disturbances or human error [13]. To incorporate these uncertainties into the model, Rong et al. [14] proposed a Gaussian Process model that represents position as a probability density function decomposed into lateral and longitudinal directions. The main limitation of these model-based approaches is that they are sensitive to dynamic specifications and model parameters. They require an explicit understanding of real-world ship dynamics and what parameter regimes are physically reasonable.

Alternatively, data-driven methods do not necessarily face the difficulties that hinder model-based approaches. Given the technological advances in the Automatic Identification System (AIS), data-driven approaches are increasingly developed to construct trajectory prediction models. Zhang et al. [15] proposed a trajectory prediction model that compressed and grouped trajectories to abstract the general motion patterns from the AIS data and use an Ant Colony algorithm to find the trajectory that conforms to the pattern. Wang et al. [16] proposed a prediction model that assigns the future trajectory calculated by a similar dynamic model to the targeting ship. However, these strategies may result in trajectories with large

differences between clustered sub-trajectories. Forti et al. [17] utilized the Long Short-Term Memory (LSTM) to predict the trajectories of the vessels, and the prediction results are verified with the AIS data in the ports of Italy. Xiao et al. [18] adopt Bidirectional LSTM to incorporate more features in trajectory data, and applied attention mechanism to improve prediction accuracy. To improve computational efficiency and avoid over-fitting, You et al. [19] proposed a Gated Recurrent Unit (GRU) network as an extended sequence-to-sequence model to predict the ship trajectory sequence for the next 5 minutes. Murray et al. [20] used auto-encoder structures to facilitate trajectory clustering to find the similar motion patterns of ships, and iteratively predict future states. Capobianco et al. [21] developed a recurrent encoder-decoder network to predict the ship trajectory with uncertainty estimation.

However, these pure data-driven models have limited interpretability. That may become a potential issue for real shipping industrial applications. Hence, model-based deep Learning [22] has been introduced by incorporating partial information derived from physics-based models. The unresolved aspects within the model are addressed through supervised deep Learning. Gao et al. [23] introduced a procedure for identifying the physical basis of the learned trajectory model, achieved by a cubic spline interpolation and an LSTM step. Xu et al. [24] introduced a physics-informed neural network (PINN), where they integrated speed and steering models into the loss function for predicting surge and sway motion of unmanned surface vehicle. Revach et al. [25] presented a recurrent neural network aided Kalman filter for partially known dynamics. This method relies on time-invariant calibrated elements for the Kalman filter, which may not adequately capture the complex process noise in ship navigation caused by factors like wind and sea effects. We address this limitation by simultaneously estimating states, coefficients, and process noise, enhancing the accuracy of ship navigation. Kanazawa et al. [26] investigated how much cooperative models, i.e., combinations of model-driven and data-driven components, benefit from the physical structure and observation data, and found that these were largely complementary, in terms of improving trajectory prediction accuracy. While the above model-based machine learning methods are verified by simulation experiments on open water areas, intersection waterways remain an open question as these areas contain different ship navigation behavior. Moreover, the Maritime Safety Administration developed the Ships' Routing System, which refined the regulations to restrict navigation behavior when ships sail to different target channels [27]. This raises complexity to such a degree that a single model may not be enough to capture the full range of motion in intersection waterways.

III. MULTIPLE MODEL OF KALMAN-GRU

A. Overview

We develop a Bayesian approach to formulate the ship trajectory prediction task, wherein the primary goal is to ascertain the subsequent predictive distribution:

$$p(\mathbf{s}_{T+1:T+\Delta T}, |, \mathbf{z}_{1:T}). \quad (1)$$

Here, $\mathbf{z}_{1:T}$ represents the observed trajectory, defined as a sequence of tuples (x_t, y_t) , $t \in [1 : T]$, where x_t and y_t denote the observed longitude and latitude measurements at time t , respectively. $\mathbf{s}_{T+1:T+\Delta T}$ denotes the predicted future state over a prediction horizon of ΔT .

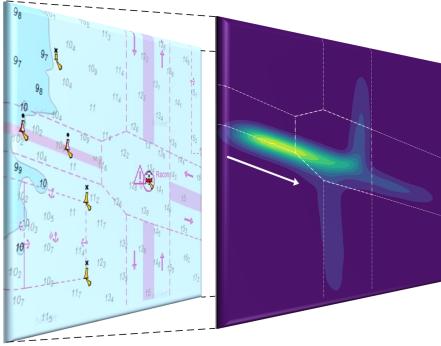


Fig. 1. Multi-mode motions in the intersection waterway. The left plot illustrates the electronic nautical chart of the South Channel in the Yangtze River Estuary, while the right plot presents the Kernel density estimation of ship trajectories within this intersection.

Existing approaches tend to utilize end-to-end deep learning methods to establish the mapping relationship between observations and predictions, typically leveraging extensive calibrated datasets. However, for intersection waterways, this can be problematic.

In this study, we used the South Channel of the Yangtze River Estuary (as illustrated in Fig. 1) as the study area. This intersecting waterway serves as a critical maritime gateway for ships navigating from the inland river in the west to diverse directional channels in the sea. It can be observed that before entering the intersection waterway, all ship trajectories are relatively concentrated, while future ship trajectories will diverge towards various directions in the waterway. The presence of such multi-mode motion patterns makes it challenging for a single end-to-end model to effectively map similar observations of ship motions to different modes in the future.

To this end, we propose a multiple Kalman-GRU model. Each Kalman-GRU is developed to capture the dynamics of a specific motion mode. Fig. 2 show the summary of the proposed model. The training trajectory data is input into the corresponding Kalman-GRU based on its calibrated mode, and the state transition coefficients and process noise are updated through network parameter learning and variational inference. Below, we discuss the details of the proposed model and describe the submodel selection procedure for the final trajectory predictions.

B. Model specification

From the AIS data, we can obtain the observations of ship trajectory $\mathbf{z}_{1:T}$. Since the AIS data contains some noise, these observations are assumed to be generated by a sequence of hidden states $\mathbf{s}_{1:T} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_T\}$. This process is formulated as a state-space model:

$$\mathbf{s}_t = \mathbf{A}\mathbf{s}_{t-1} + \epsilon_t^s, \quad \mathbf{z}_t = \mathbf{B}\mathbf{s}_t + \epsilon_t^o, \quad (2)$$

where \mathbf{A} and \mathbf{B} are the state transition and measurement matrices, respectively, and $\epsilon_t^s \sim \mathcal{N}(\epsilon_t^s | 0, \mathbf{Q})$ denotes process noise with covariance matrix \mathbf{Q} while $\epsilon_t^o \sim \mathcal{N}(\epsilon_t^o | 0, \mathbf{R})$ denotes measurement noise with covariance matrix \mathbf{R} .

The process noise covariance matrix \mathbf{Q} describes how, on average, the state transition deviates from the deterministic dynamics. Standard Kalman filters require \mathbf{Q} to be fixed. However, ship dynamics are subject to uncertainty arising from environmental conditions and operational factors. Moreover, these uncertain dynamics may vary across different modes within the intersection waterway. As such, we believe that the quality of a covariance matrix, once calibrated and fixed, will deteriorate over time [28]. We therefore propose an online estimator which updates the process noise covariance matrix after every ship trace measurement. For numerical stability during inference, we will estimate the inverse of covariance matrix, the precision matrix: $\Lambda = \mathbf{Q}^{-1}$. Our prior for Λ thus becomes:

$$p(\Lambda) \sim \mathcal{W}(\Lambda | \mathbf{V}_0, d_0) \quad (3)$$

where \mathbf{V}_0 is the scale matrix and d_0 are the degrees of freedom of the Wishart distribution. Note that the conjugate prior for a precision matrix in a Gaussian distribution is the Wishart distribution [29], which motivates the choice in Equation (3).

Traditionally, the state transition matrix \mathbf{A} in a Kalman filter is designed by an expert based on physical knowledge of a specific ship's dynamics. However, a fixed transition matrix ignores changes in ship dynamics over time, such as maneuvers to turn the ship towards the target channel, and may lead to poor trajectory prediction performance. We address this problem by introducing time-varying transition dynamics driven by a neural network, referred to as a K-GRU model. Incorporating the Gaussian distribution of the process noise, the state transition may be specified as:

$$p(\mathbf{s}_t | \mathbf{s}_{t-1}, \Lambda; \theta) = \mathcal{N}(\mathbf{s}_t | \mathbf{A}_t \mathbf{s}_{t-1}, \Lambda^{-1}). \quad (4)$$

In this equation, “;” is used to separate parameters, indicating that θ are parameters of the probability model, consistent with the notation in [29]. The expression $\mathcal{N}(\mathbf{s}_t | \mathbf{A}_t \mathbf{s}_{t-1}, \Lambda^{-1})$ denotes a multivariate Gaussian distribution with mean $\mathbf{A}_t \mathbf{s}_{t-1}$ and covariance matrix Λ^{-1} . Moreover, $\mathbf{A}_t[j, j] = \mathbf{a}_t[j]$ represents that the j -th diagonal element of \mathbf{A} corresponds to the j -th element of vector \mathbf{a}_t . These \mathbf{a}_t are driven by the GRU network Φ with parameters θ : $\mathbf{a}_t = \Phi(\mathbf{z}_{1:t-1}; \theta)$. Consequently, the transition matrix \mathbf{A}_t approximates the ship motion dynamics at each time step. This model can be regarded as a data-driven Extended Kalman Filter, where the approximation is informed by historical observations.

Due to the Gaussian assumption on the measurement noise, the likelihood is also Gaussian distributed:

$$p(\mathbf{z}_t | \mathbf{s}_t) = \mathcal{N}(\mathbf{z}_t | \mathbf{B}\mathbf{s}_t, \mathbf{R}). \quad (5)$$

The measurement noise covariance matrix \mathbf{R} is assumed to be time-invariant and can be independently determined by the AIS data collection protocol.

Lastly, we define a prior distribution on the states \mathbf{s}_0 :

$$p(\mathbf{s}_0) = \mathcal{N}(\mathbf{s}_0 | \mathbf{m}_0, \mathbf{P}_0). \quad (6)$$

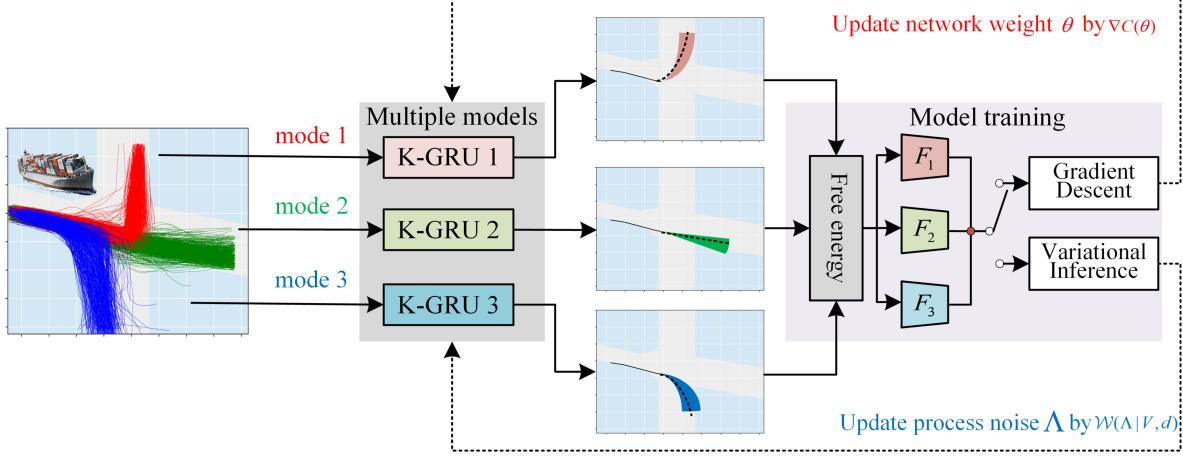


Fig. 2. Training process of multiple models of K-GRU. The training samples are feed to the specific K-GRU corresponding to the modes calibrated by the target channel. Through learning and inference, the network parameter and process noise are optimized.

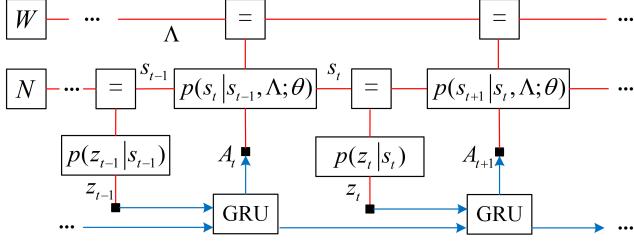


Fig. 3. A factor graph for K-GRU in (7). Edges represent the variables associated with each node function. For example, the node $p(z_t | s_t, \Lambda; \theta)$ is associated with two variables: z_{t-1} and s_{t-1} . The red edges are the variables in Kalman filer, the blue arrows are the variables in GRU. Observation z_{t-1} are fed into GRU, and generate the transition matrix A_t for Kalman filter. The triple dots indicate a graph continuation in temporal directions.

Given the above specifications, we may visualize it as a factor graph shown in Fig. 3 and write the full model specification of one K-GRU for a sequence of length T as:

$$p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \theta) = \underbrace{p(\Lambda)p(\mathbf{s}_0)}_{\text{priors}} \prod_{t=1}^T \underbrace{p(\mathbf{s}_t | \mathbf{s}_{t-1}, \Lambda; \theta)}_{\text{state transition}} \underbrace{p(\mathbf{z}_t | \mathbf{s}_t)}_{\text{likelihood}}. \quad (7)$$

C. Learning and inference

Training a K-GRU means learning the optimal GRU network parameters θ and inferring the most likely process noise precision Λ . Simultaneous learning and inference can be achieved by maximizing the log-evidence $\log p(\mathbf{z}_{1:T})$. However, evaluating the log-evidence requires marginalizing over both $\mathbf{s}_{1:T}$ and Λ , which is intractable. To this end, we define an upper bound \mathcal{F} to the log-evidence, referred to as a free energy functional [30]:

$$\mathcal{F}[q] = \mathbb{E}_q \left[\log \frac{q(\mathbf{s}_{0:T}, \Lambda)}{p(\mathbf{s}_{0:T}, \Lambda | \mathbf{z}_{1:T}; \theta)} \right] - \log p(\mathbf{z}_{1:T}). \quad (8)$$

It's worth noting that the negative free energy is also known as the Evidence Lower Bound (ELBO) [31], which serves as a key objective function in variational inference. \mathbb{E}_q denotes the

expectation operator with respect to the variational distribution $q(\mathbf{s}_{1:T}, \Lambda)$. This variational distribution approximates the posterior distribution over the unknown variables in the model. Evaluating this objective requires the posterior distribution of interest, which means this form cannot be used. To proceed, we absorb the model evidence term, yielding

$$\mathcal{F}[q] = \mathbb{E}_q \left[\log \frac{q(\mathbf{s}_{0:T}, \Lambda)}{p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \theta)} \right], \quad (9)$$

which *can* be evaluated because the joint distribution factorizes into known terms, i.e., the priors, state transitions and likelihoods (7). we introduce a mean-field factorization for the variational distribution:

$$q(\mathbf{s}_{0:T}, \Lambda) = q(\mathbf{s}_{0:T}) \cdot q(\Lambda). \quad (10)$$

This factorization separates the joint distribution into independent factors, facilitating more tractable optimization. Although the mean-field assumption may neglect certain dependencies within ship dynamics, thereby potentially resulting in inaccuracies in inference outcomes, in scenarios necessitating real-time ship trajectory prediction, the trade-off of sacrificing a modest degree of accuracy for enhanced computational efficiency may be reasonable. We will occasionally adopt q_s and q_Λ as short-hand notation for $q(\mathbf{s}_{0:T})$ and $q(\Lambda)$. Inference corresponds to finding the optimal forms of each of the variational distributions, i.e., the forms that minimize the free energy functional specified in (9). These optimal forms can be derived analytically (see [30] for details on the procedure):

$$q(\mathbf{s}_{0:T}) \propto \exp(\mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})]) \quad (11)$$

$$q(\Lambda) \propto \exp(\mathbb{E}_{q_s} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})]). \quad (12)$$

Computing the expectation in (11) gives: (see Appendix for detailed derivation):

$$\exp(\mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})]) \propto \prod_{t=0}^T \mathcal{N}(\mathbf{s}_t | \mathbf{m}_t, \mathbf{P}_t). \quad (13)$$

The parameters of each Gaussian state distribution follow a typical prediction-correction procedure. For $t > 0$, the

predictions for \mathbf{s}_t are given by the Chapman-Kolmogorov equation:

$$p(\mathbf{s}_t | \mathbf{z}_{1:t-1}) = \int p(\mathbf{s}_t | \mathbf{s}_{t-1})p(\mathbf{s}_{t-1} | \mathbf{z}_{1:t-1})d\mathbf{s}_{t-1}. \quad (14)$$

Then the mean \mathbf{m}_t^- and covariance \mathbf{P}_t^- of predictions for \mathbf{s}_t can be calculated:

$$\mathbf{m}_t^- = \mathbf{A}_t \mathbf{m}_{t-1}, \quad \mathbf{P}_t^- = \mathbf{A}_t \mathbf{P}_{t-1} \mathbf{A}_t^\top + (d\mathbf{V})^{-1}. \quad (15)$$

The \mathbf{V} and d are parameters of the optimal $q(\Lambda)$, which will be explained shortly hereafter. The state predictions are corrected by information from observations through the Bayes' rule:

$$p(\mathbf{s}_t | \mathbf{z}_{1:t}) = \frac{1}{Z_t} p(\mathbf{z}_t | \mathbf{s}_t) p(\mathbf{s}_t | \mathbf{z}_{1:t-1}) \quad (16)$$

where Z_t represents the normalization constant $Z_t = p(\mathbf{z}_t | \mathbf{z}_{1:t-1})$. Then, the updated result for (13) is given by

$$\mathbf{m}_t = \mathbf{m}_t^- + \mathbf{P}_t^- \mathbf{B}^\top (\mathbf{B} \mathbf{P}_t^- \mathbf{B}^\top + \mathbf{R})^{-1} (\mathbf{z}_t - \mathbf{B} \mathbf{m}_t^-) \quad (17)$$

$$\mathbf{P}_t = \mathbf{P}_t^- - \mathbf{P}_t^- \mathbf{B}^\top (\mathbf{B} \mathbf{P}_t^- \mathbf{B}^\top + \mathbf{R})^{-1} \mathbf{B} \mathbf{P}_t^- \top. \quad (18)$$

Since there is no observation for the prior state \mathbf{s}_0 , it is not updated and thus $q(\mathbf{s}_0) = p(\mathbf{s}_0)$.

The optimal form of the process noise precision is proportional to a Wishart distribution, (see the Appendix for details on the derivation),

$$\exp(\mathbb{E}_{q_s} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})]) \propto \mathcal{W}(\Lambda | \mathbf{V}, d), \quad (19)$$

with $d = d_0 + T$ degrees of freedom and

$$\begin{aligned} \mathbf{V}^{-1} &= \mathbf{V}_0^{-1} + \sum_{t=1}^T (\mathbf{P}_t - \mathbf{A} \mathbf{P}_{t-1} - \mathbf{P}_{t-1} \mathbf{A}^\top + \mathbf{P}_{t-1} \\ &\quad + (\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1})(\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1})^\top)^{-1}. \end{aligned} \quad (20)$$

The parameter \mathbf{V} above depends on the parameters of the state posteriors, \mathbf{m}_t and \mathbf{P}_t (Eqs. 17, 18). At the same time, the state parameters \mathbf{m}_t and \mathbf{P}_t depend on \mathbf{V} and d (15). The simultaneous dependence of the parameters implies the solution requires iteration. This variational inference procedure is, in fact, a coordinate descent algorithm, for which convergence is guaranteed because each update only produces an equal or smaller value of the objective function (9) [32].

For learning the network parameters θ , we drop terms inside the logarithm in (9) that do not involve θ and replace the expectation with respect to $q(\mathbf{s}_{1:T})$ with a marginalization:

$$\mathcal{C}(\theta) = - \int \mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{1:T}, \Lambda; \theta)] d\mathbf{s}_{1:T}. \quad (21)$$

It can be seen that there are mutual dependencies between the network parameters θ and the model parameters $\mathbf{s}_{0:T}$ and Λ . On one hand, appropriate values of θ enhance prediction accuracy in 15, minimize the disparity between predictions and observations, and yield more precise inference outcomes for $q(\mathbf{s}_{0:T})$ and $q(\Lambda)$ in Kalman component. On the other hand, precise inference outcomes for $q(\mathbf{s}_{0:T})$ and $q(\Lambda)$ reveal the hidden ship dynamics with uncertainty, thereby facilitating the GRU component in capturing state transitions. Fortunately, these mutual dependencies can be addressed through the same objective function 9. Specifically, we perform iterative learning

and inference on the entire training data set for a period of 100 epochs. During each epoch, we update the parameters for $\mathbf{s}_{0:T}$ and Λ using 13 and 19, and optimize the network parameters θ using the gradient of $\mathcal{C}(\theta)$. This procedure is outlined in Algorithm 1.

Algorithm 1 Learning and inference for K-GRU

Input: Training data set, initial GRU network parameters θ , parameters of prior state $\mathbf{m}_0, \mathbf{P}_0$, parameters of prior precision \mathbf{V}_0, d_0 .

- 1: **for** $n = 1$ to N **do**
- 2: Sample a trajectory $\mathbf{z}_{1:T}$ from training data set
- 3: Update the state estimates $q(\mathbf{s}_{0:T})$ by (13)
- 4: Update the precision estimate $q(\Lambda)$ by (19)
- 5: Update the parameters θ by gradient descent on $\mathcal{C}(\theta)$
- 6: **end for**

Output: $q(\mathbf{s}_{0:T}), q(\Lambda), \theta$

Compared to the conventional GRU, the integration of inference and learning in Algorithm 1 enhances trajectory prediction by incorporating uncertainty. These predictions not only yield deterministic paths but also provide reachable areas with confidence levels, crucial for ship collision avoidance. Additionally, the inference component can regulate network training and prevent over-fitting. For instance, in the case of an anomalous trajectory, it is feasible to infer a smaller precision noise Λ , resulting in a relatively minor impact on $C(\theta)$ compared to the traditional Mean Squared Error (MSE) loss function.

D. Online model selection

During training, a K-GRU model is fit to each mode of the ship's motion. In order to make a single trajectory prediction, we select the K-GRU that best predicts the ship's most recent position measurements and extend that K-GRU to future time points. Fig. 4 provides an overview of the submodel selection procedure. Let t_0 be the time point of the initial measurement in our time-series and t_2 be the current time point. We split the observations from t_0 to t_2 based on a look back time t_b , thereby producing a set of older observations $\mathbf{z}_{t_0:t_1}$ and a set of more recent observations $\mathbf{z}_{t_1+1:t_2}$, where $t_1 = t_2 - t_b$. Using the older observations, each K-GRU submodel makes a prediction for the newer observations using (7).

Each proposal is evaluated on the recent observations $\mathbf{z}_{t_1+1:t_2}$ in terms of a predictive likelihood (PL):

$$PL_{t_2}^{(m)} = \prod_{t=t_1+1}^{t_2} \mathcal{N}(\mathbf{z}_t | \mu_t^{(m)}, \Sigma_t^{(m)}), \quad (22)$$

where $\mu_t^{(m)}, \Sigma_t^{(m)}$ are the mean and covariance of predicted distribution by m-th GRU at time t . Note that this function evaluates the probability of these observations under the predictive distribution, which is computed iteratively by aggregating single-step predictions. Specifically, previous predictions are leveraged as observations to forecast future states. In this context, the transition matrix \mathbf{A}_{t_3} computed by the GRU model is denoted as $\mathbf{A}_{t_3} = \Phi([\mathbf{z}_{1:t-1}, \mu_{t:t-1}])$, where

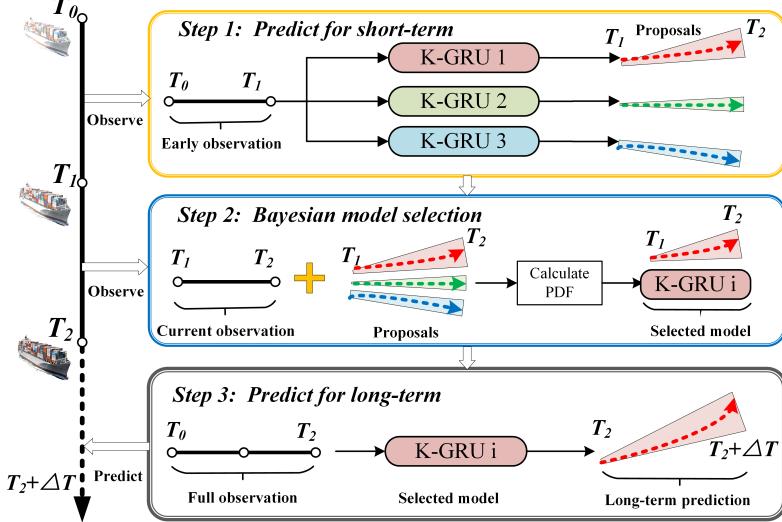


Fig. 4. Online model strategy for the MK-GRU model. The observations from t_0 to t_2 are divided into older observations $\mathbf{z}_{t_0:t_1}$ and newer observations $\mathbf{z}_{t_1:t_2}$. Each submodel is fit to the older data and makes a prediction for the newer data. The one with the most accurate prediction (22) is selected and used to create a single trajectory prediction for $t_2 + \Delta T$.

$[\mathbf{z}_{1:t-1}, \mu_{t:t_3-1}]$ represents the concatenated vector comprising the observations and the mean of the predictive distributions. As such, $PL_{t_2}^{(m)}$ represents a *score* for the m -th K-GRU. The K-GRU with the maximal score, i.e., the one with the highest total probability under the prediction, is selected and used to make a final trajectory prediction for the future, from t_2 to the end of the time horizon t_e . This procedure is outlined in Algorithm 2.

Algorithm 2 Online model selection

Input: Current time t_2 , look back time t_b , predict horizon ΔT , and end time t_e

- 1: **repeat**
- 2: $t_1 = t_2 - t_b$
- 3: **for** $m = 1$ to M **do**
- 4: Update $PL_{t_2}^{(m)}$ by (22)
- 5: **end for**
- 6: $m^* = \arg \max PL_{t_2}^{(m)}$
- 7: Update $p(\mathbf{s}_{t_2+1:t_2+\Delta T})$ by (7) with m^* -th GRU
- 8: **until** $t_2 + \Delta T > t_e$

Output: $p(\mathbf{s}_{t_2+1:t_2+\Delta T})$

In Algorithm 2, a lookback time t_b controls the duration for which observations influence the model selection process. This differs from the conventional Bayesian multiple model [33], which recursively uses all observations to estimate hidden mode probabilities. The flexibility of t_b allows captains to customize the model's inference results to their needs. A longer lookback time can produce more robust and consistent mode inference results but may delay the identification of hidden modes. Conversely, a shorter t_b , such as 1, enables the model to respond promptly to modes but may increase the risk of misjudgments.

IV. EXPERIMENTS

We discuss experiments and results in both a quantitative analysis as well as a qualitative analysis (case studies).

A. Data set

We conducted experiments using both real-world AIS data and synthetic data. The AIS dataset comprises ship trajectories within the South Channel of the Yangtze River Estuary, extending longitudinally from $122^{\circ}25'E$ to $122^{\circ}37'E$ and latitudinally from $30^{\circ}48'N$ to $31^{\circ}6'N$. This intersection serves as a critical gateway for ships traveling from the inland river to sea channels in various directions.

TABLE I
AIS TRAJECTORY DATA SET WITH MULTI-MODE LABELS.

Mode type	Training samples	Testing samples
mode 1 (to North)	50	1100
mode 2 (to East)	50	850
mode 3 (to South)	50	339

TABLE II
SYNTHETIC TRAJECTORIES DATA SET WITH VARYING NUMBERS OF MODES.

Mode numbers	Training samples for each mode	Testing samples
1 mode	120	200
2 modes	60	200
3 modes	40	200
4 modes	30	200
5 modes	24	200

Note that AIS data inherently manifests asynchronous. Even in densely trafficked intersection waterways with elevated broadcast frequencies, the AIS broadcast frequency varies among different ships, and data gaps may arise for the same ship due to various factors. Hence, we employed an interpolation and resampling approach as outlined in [34] to obtain regularly-sampled trajectories. Firstly, we extract

trajectory data from various ships using Maritime Mobile Service Identity (MMSI) and timestamp information. Next, we employ a sliding window technique to detect trajectory sparsity. If a trajectory contains a significant amount of missing data, it is discarded as it may not accurately record the ship's movements. For trajectories with a limited amount of missing data, Hermitian cubic interpolation is applied to fill in the gaps and generate trajectories sampled at 1 Hz. Subsequently, the trajectories are resampled to a frequency of 1/18 Hz (one point recorded at intervals of 18 seconds.).

Following the interpolation and resampling process, the AIS dataset utilized in this study comprised 2439 ship trajectories. Each trajectory was assigned a motion modality label based on its destination (North, East, or South). To construct the training set, we randomly selected 50 trajectories from each label class. That left 1100 test samples for the North destination, 850 for East and 339 samples for South.

Moreover, synthetic data was employed to evaluate the efficacy of the MK-GRU model, with particular focus on discussing the impact of varying numbers of modes on its performance. Following the methodology presented in [35], a total of 600 trajectories, each lasting 240 seconds, were generated and resampled at 1/2 Hz, yielding 120 sampling points per trajectory. Initial coordinates (x and y) were fixed at 0, and both thruster angle and revolution remained at zero until the 50th second. The initial surge velocity (in meters per second) followed a Gaussian distribution $\mathcal{N}(4, 1)$. Subsequently, after a random time T_m seconds, thruster revolution (in RPM) was set to n_{max} and thruster angle (in degrees) to δ_{max} . Here, T_m was uniformly sampled from the range [50, 70], n_{max} was drawn from $\mathcal{N}(5, 0.8)$, and δ_{max} was drawn from $[-35, 35]$. The trajectories selected for the synthetic dataset occur after the 50-second mark. These ship control simulations were conducted using the Julia package ShipMMG [36].

Unlike the real-world AIS trajectories in the South Channel, which naturally segregate into three modes based on their target channels, the synthetic trajectories manifest in a more continuous space. This characteristic provides the opportunity to adjust the number of motion modes. Employing the Dynamic Time Warping (DTW) method and K-means clustering [37], the synthetic trajectories are partitioned into K clusters, with K ranging from 2 to 5 in this study. Fig. 5 illustrates the cluster results obtained when the number of modes is set to 2 and 5. The training data are sampled proportionally according to the number of modes, ensuring a total of 120 training samples. Additionally, 200 trajectories are sampled for the test data. The details of the training and test data for synthetic trajectories are provided in Table II.

B. Setup and results

The MK-GRU model is implemented using the Julia package RxInfer [38]. For comparison, we implemented several baseline models: a pure GRU network [39], an LSTM-MDN network [40], and a Seq2Seq model [19]. We also compare it against a single K-GRU model, allowing us to demonstrate the usefulness of the multiple model approach.

As performance metrics, we employ the average displacement error (ADE) and the final displacement error (FDE).

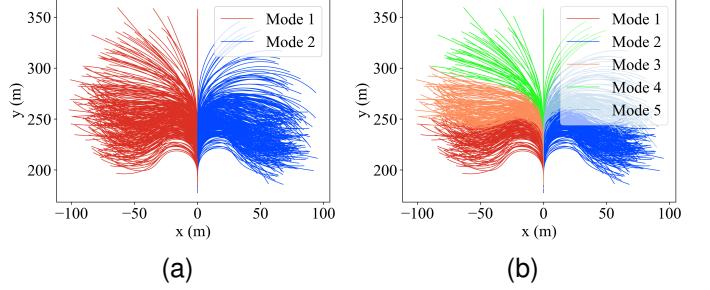


Fig. 5. Multi-modes in the synthetic ship trajectories. (a) Illustrating the trajectories when the number of modes is set to 2. (b) Illustrating the trajectories with the number of modes set to 5.

Instead of employing Euclidean distance, we utilize geographical distance metrics for latitude and longitude, as done in previous work [41]. Specifically, ADE measures the average geographical distance between predicted and observed positions, while FDE quantifies the geographical distance between the predicted and observed positions at the final time point within the prediction horizon. Table III compares the ADE and FDE of various methods, given different observation lengths and prediction horizons. For instance, "9-18" represents using a 9-minute input trajectory to forecast the next 18-minute. The ADE metric demonstrates that the MK-GRU outperforms the other methods for all combinations of observation length and prediction horizon. For the FDE metric, the MK-GRU achieved the best performance only in the 9-18 setting, while LSTM-MDN and Seq2seq performed the best in the 13.5-13.5 and 18-9 groups, respectively. The MK-GRU model performed competitively (underlined values indicating the second best performance). These results are likely due to the accumulation of prediction errors in the recursive computations in GRU/K-GRU/MK-GRU models moving forward in time. In contrast, LSTM-MDN directly outputs the entire future trajectory through the last dense layer, which mitigates error accumulation. However, this approach sacrifices the temporal dependency in its predictions, resulting in frequent trajectory oscillations that are not consistent with the actual movements of ships.

To assess the plausibility of the predicted trajectories, we introduced a metric called the Difference-of-Course-Over-Ground (DCOG), which measures the COG at each time-step between the predicted trajectory and the ground-truth trajectory.

A higher DCOG value indicates that the predicted trajectory is more likely to display frequent changes in COG compared to the actual trajectory, which is inconsistent with the continuous movement patterns of ships. Table IV shows the DCOG values of various methods for different observation lengths and prediction horizons. It can be seen that the MK-GRU outperforms the other methods in the 9-18 and 13.5-13.5 groups, while the K-GRU performs better in the 18-9 group. Notably, the DCOG value for LSTM-MDN ranges between 38 and 52 degree, which is highly unrealistic as it implies a significant change in course over a period of 18 seconds.

TABLE III
PREDICTION ERROR (KM) AS A FUNCTION OF OBSERVATION LENGTH AND PREDICTION HORIZON (MINUTES).

Obs-Pre Method \ Metrics (km)	9-18		13.5-13.5		18-9	
	ADE	FDE	ADE	FDE	ADE	FDE
GRU	1.527 (0.656)	4.767 (2.853)	0.918 (0.847)	3.746 (2.702)	0.933 (0.855)	2.296 (2.246)
LSTM-MDN	1.212 (0.987)	2.847 (0.745)	0.333 (0.229)	0.571 (0.551)	0.443 (0.335)	0.940 (0.454)
Seq2seq	1.172 (0.920)	2.836 (0.594)	0.223 (0.262)	0.754 (0.406)	0.338 (0.230)	0.592 (0.474)
K-GRU	1.209 (1.093)	3.286 (1.307)	0.635 (0.550)	1.125 (1.066)	0.397 (0.348)	0.976 (1.014)
MK-GRU	0.886 (0.875)	2.418 (1.848)	0.217 (0.193)	<u>0.771</u> (0.670)	0.323 (0.534)	<u>0.678</u> (0.744)

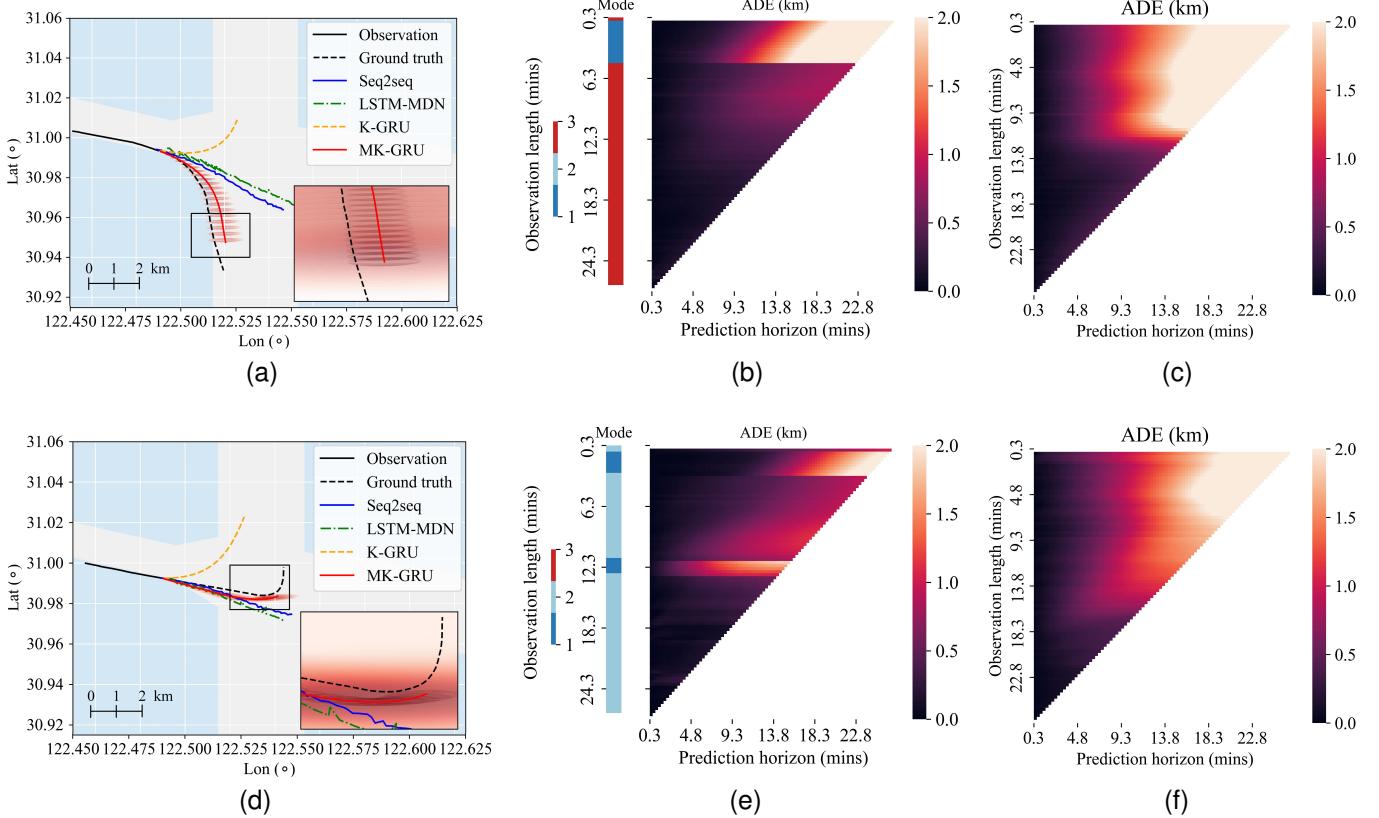


Fig. 6. Prediction in two cases and error heatmap for observation length and prediction horizon. (a) Predict 18-minute in the future by 9-minute observations for Case 1. (b) Mode selection and error heatmap for MK-GRU in Case 1, (c) error heatmap for K-GRU for Case 1, (d) Predict 18-minute in the future by 9-minute observations for Case 2. (e) Mode selection and error heatmap for MK-GRU in Case 2, (f) error heatmap for K-GRU for Case 2.

TABLE IV
PLAUSIBILITY OF PREDICTED TRAJECTORY (DCOG °).

Obs-Pre Method \ Metrics	9-18	13.5-13.5	18-9
GRU	1.723 (1.242)	1.727 (2.569)	1.318 (2.461)
LSTM-MDN	38.637 (45.933)	62.937 (66.015)	52.210 (76.609)
Seq2seq	49.831 (46.653)	10.049 (11.305)	2.546 (1.786)
K-GRU	0.773 (1.251)	0.985 (0.947)	1.078 (0.965)
MK-GRU	0.603 (0.616)	0.797 (0.685)	1.198 (0.791)

C. Case studies

To visualize the accuracy and plausibility of our model, we selected a few subsets of the test data and performed qualitative analyses (code as well as animations are available

online¹). Case 1 focuses on a ship traveling from an inland channel to the South channel, and case 2 depicts a ship traveling to the North channel. Fig. 6a illustrates the true and predicted trajectories generated by each model for predicting 18 minutes into the future based on 9-minute observations. The baseline methods and single K-GRU predict the wrong direction entirely while the MK-GRU correctly predicts the path towards the South channel.

Case 2 (Fig. 6d) shows a similar situation as in Case 1, with the MK-GRU outperforming other methods in terms of accuracy. From the zoomed-in portion of the figure, it can be observed that LSTM-MDN predicts erratic trajectories (i.e., high DCOG). Note that the color in the zoomed-in portion

¹github.com/Chengfeng-Jia/ship-trajectory-prediction-MKGRU.

represents the uncertainty of predicted distribution of MK-GRU.

It can be seen the multiple model was capable of inferring the ship's intended course towards the South Channel at an earlier stage, even before the actual turning maneuver occurred. On the other hand, the single K-GRU model misjudged the future direction. In case 2, the single K-GRU predicted the right direction, but still produced a significant deviation from the actual trajectory. Capturing ship dynamics with multiple modalities is challenging for a single model. However, in a multiple model, each sub-model can be viewed as an expert specializing in a particular mode, resulting in more accurate predictions within their respective areas of expertise.

Then, we plot the error heatmaps that illustrate the impact of varying the observation length and prediction horizon on the model's performance. The bars in Fig. 6b and Fig. 6e on the left represent the model selection results given different observation lengths. The heatmap in Fig. 6b shows the prediction error of the MK-GRU model, with the horizontal and vertical axes representing the prediction horizon and the observation length, respectively. The darker the color, the smaller the error. The heatmap reveals a sudden change in the error at around 4.5 minutes of observation, which coincides with the moment when the multiple model selected the correct direction mode. On the other hand, the prediction error of the single K-GRU model in Fig. 6c remains high for longer, indicating the value of the multiple model approach. The same trend is observed in Fig. 6e and 6f, with the exception of two changes in model selection corresponding to two sudden changes in prediction error.

D. Impact of multiple model

To illustrate the relationships between observations, motion mode, and real-time predictions, we design a ship navigation dashboard. This dashboard has the potential to provide captains with valuable insights into the future movements and trends of surrounding ships, thereby enhancing their situational awareness capabilities. Fig. 7 and Fig. 8 showcase the ship navigation dashboard for the same ship at different time steps. Fig. 7 (a) displays the initial observation at the 0.6 minutes. Fig. 7 (b) presents the probabilities of different motion modes based on limited observations, indicating that mode 1 is dominant. Fig. 7 (c) provides the rationale behind this, as the prediction from K-GRU 1 closely matches the ground truth. Subsequently, Fig. 7 (d) illustrates the long-term prediction generated by K-GRU 1. Fig. 8 (a) displays the updated observation at the 4.8 minutes. Fig. 7 (b) illustrates the predominance of mode 3, as evident from Fig. 7 (c), where K-GRU 3 yields the most accurate prediction. The identification of the correct mode enhances the precision of the long-term prediction shown in Fig. 7 (d).

We analyze the relationship between model selection error (ME) and prediction error (PE) across the AIS test data set. Fig. 9 shows ME and PE for the three modalities as a function of observation length. There is a certain correlation between PE and ME, particularly when the observation length is sufficient (when the observation length exceeds 9 minutes).

It is observed that both PE and ME decrease simultaneously given sufficient observations. This suggests that accurately identifying the ship's specific direction modality may improve the accuracy of trajectory prediction.

To investigate the effect of varying the number of modes on the multi-model, the number was systematically adjusted from 1 to 5. Fig. 10 illustrates the resultant variation of ADE corresponding to the mode number. Fig. 10a depicts the ADE outcomes for a 90-second observation window, forecasting 90 seconds into the future. It is evident that configuring the number of modes to 2 and 3 yields superior performance compared to a single-mode setting. However, as the number of modes increases (4 and 5), the error also escalates, exceeding that of the single-mode configuration. Fig. 10b depicts the ADE outcomes for a 120-second observation window, forecasting 60 seconds into the future. It can be seen that setting the number of modes to 4 and 5 does not yield significant differences in mean ADE values compared to when set to 1. Fig. 10 indicates that increasing the number of modes does not inherently enhance performance; rather, an excess of modes may lead to decreased predictive accuracy. Additionally, the computational workload for training and data calibration substantially increased with a higher number of modes.

V. DISCUSSION

Our findings suggest that a multiple model can make accurate predictions, but that its performance depends heavily on accurate and timely model selection. Although our proposed method predicts turning trends well, in some cases, deviations from the final point of the prediction horizon were observed. This may be attributed to our definition of motion modality, which only relies on target direction and does not consider acceleration. Furthermore, it is notable that this study employs spherical coordinates (latitude and longitude) directly in the Kalman-GRU for the AIS data. This may potentially introduce additional errors into the ship dynamic model. Nevertheless, our method remains a technical advancement for ship navigation, as it not only provides predicted trajectories but also quantifies uncertainty. This can be a crucial reference for captains to accommodate situational awareness and assist the decision-making process in ship avoidance scenarios.

VI. CONCLUSION

We proposed a probabilistic ship trajectory prediction method based on multiple Kalman-GRU models that address the challenges posed by the time-varying dynamics and the multi-mode nature of ship motions. Data from both real-world ship trajectories and simulated trajectories were collected and utilized for model validation. Quantitative metrics and case studies. Our method outperformed a set of competitive baseline models in most metrics. Our findings suggest that correctly identifying the specific motion mode of a ship is critical for prediction accuracy. The outcomes of this study are anticipated to yield precise short-term trajectory predictions, with potential applications in situational awareness and collision avoidance.

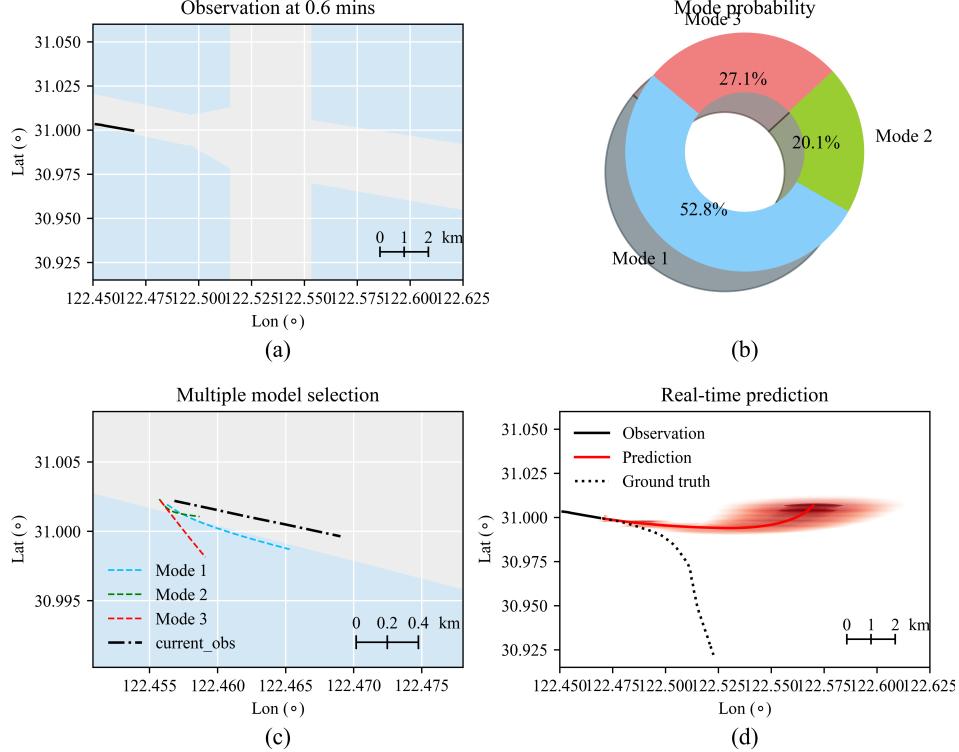


Fig. 7. Ship navigation dashboard at the 0.6 minutes. (a) Updated trajectory observation, (b) Probability of motion modes (c) Multiple model selection, (d) Real-time prediction with uncertainty.

The proposed method relies on regularly-sampled observations, which may limit its accuracy in practice, particularly for AIS data that exhibit asynchronous characteristics. As a potential future work, we aim to extend the MK-GRU framework to incorporate real-time reactive learning and prediction capabilities. This extension will enable the model to adapt to missing data or asynchronous trajectories commonly observed in real-world AIS data.

APPENDIX

Here, we treat the derivation of the optimal form the variational distribution of the states (11) in more detail, starting with the expectation over $q(\Lambda)$. Using (7) and the linearity property of expectations, we get:

$$\begin{aligned} \mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})] &= \mathbb{E}_{q_\Lambda} [\log p(\Lambda)] + \mathbb{E}_{q_\Lambda} [\log p(\mathbf{s}_0)] \\ &+ \sum_{t=1}^T \mathbb{E}_{q_\Lambda} [\log p(\mathbf{s}_t | \mathbf{s}_{t-1}, \Lambda; \hat{\theta})] + \mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_t | \mathbf{s}_t)]. \end{aligned} \quad (23)$$

The expectations over terms that don't involve Λ drop out:

$$\mathbb{E}_{q_\Lambda} [\log p(\mathbf{s}_0)] = \log p(\mathbf{s}_0) \quad (24)$$

$$\mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_t | \mathbf{s}_t)] = \log p(\mathbf{z}_t | \mathbf{s}_t). \quad (25)$$

Furthermore, the variational distribution $q(\mathbf{s}_{0:T})$ is a function over states, which means any term not involving the states acts only as a normalization. Specifically, $\mathbb{E}_{q_\Lambda} [\log p(\Lambda)]$ does

not involve any \mathbf{s}_t and may be ignored here. That leaves the expectation over the state transition:

$$\begin{aligned} \mathbb{E}_{q_\Lambda} [\log p(\mathbf{s}_t | \mathbf{s}_{t-1}, \Lambda; \hat{\theta})] &= -\log(2\pi) + \mathbb{E}_{q_\Lambda} \left[\frac{1}{2} \log |\Lambda| \right] \\ &- \mathbb{E}_{q_\Lambda} \left[\frac{1}{2} (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1})^\top \Lambda (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1}) \right]. \end{aligned} \quad (26)$$

The second term on the right-hand side above does not depend on states, so it may be ignored. The precision matrix Λ only appears linearly in the quadratic term and we may thus use the linearity property to replace it with the mean of $q(\Lambda)$:

$$\begin{aligned} \mathbb{E}_{q_\Lambda} \left[\frac{1}{2} (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1})^\top \Lambda (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1}) \right] &= \\ \frac{1}{2} (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1})^\top (d\mathbf{V}) (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1}). \end{aligned} \quad (27)$$

Putting it all back together, we have:

$$\begin{aligned} \mathbb{E}_{q_\Lambda} [\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \Lambda; \hat{\theta})] &= \\ -\frac{1}{2} (\mathbf{s}_0 - \mathbf{m}_0)^\top \mathbf{P}_0^{-1} (\mathbf{s}_0 - \mathbf{m}_0) &- \sum_{t=1}^T \frac{1}{2} (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1})^\top (d\mathbf{V}) (\mathbf{s}_t - \mathbf{A}_t \mathbf{s}_{t-1}) \\ - \frac{1}{2} (\mathbf{z}_t - \mathbf{B} \mathbf{s}_t)^\top \mathbf{R}^{-1} (\mathbf{z}_t - \mathbf{B} \mathbf{s}_t) + C, \end{aligned} \quad (28)$$

where C represents constant terms, i.e., terms that do not depend on any \mathbf{s}_t . Exponentiating the above equation yields a

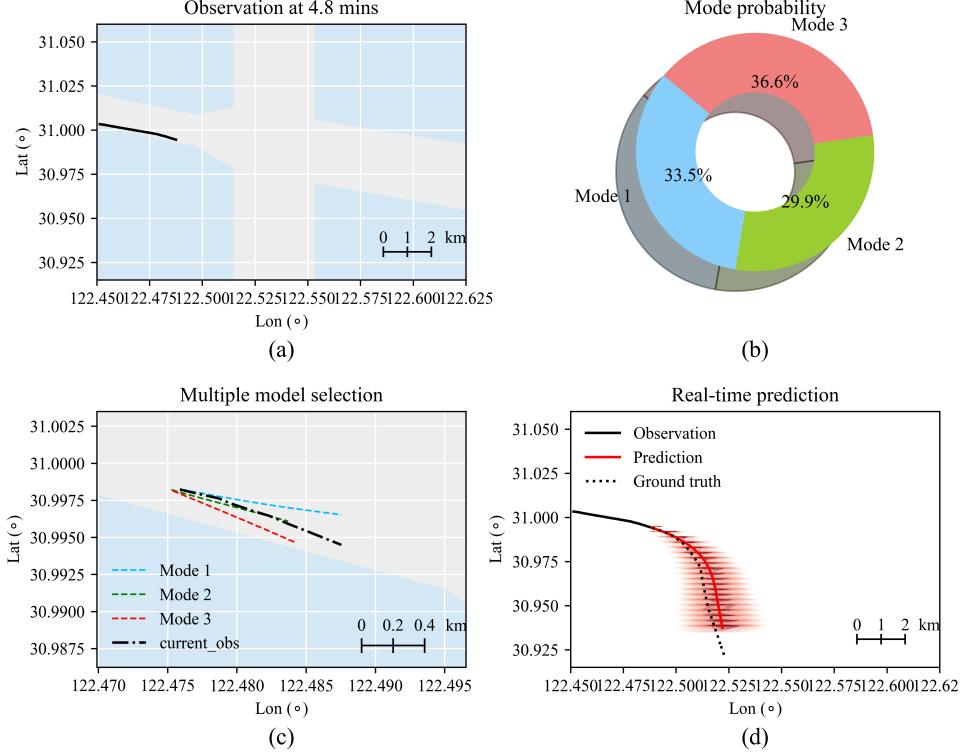


Fig. 8. Ship navigation dashboard at the 4.8 minutes. (a) Updated trajectory observation, (b) Probability of motion modes (c) Multiple model selection, (d) Real-time prediction with uncertainty.

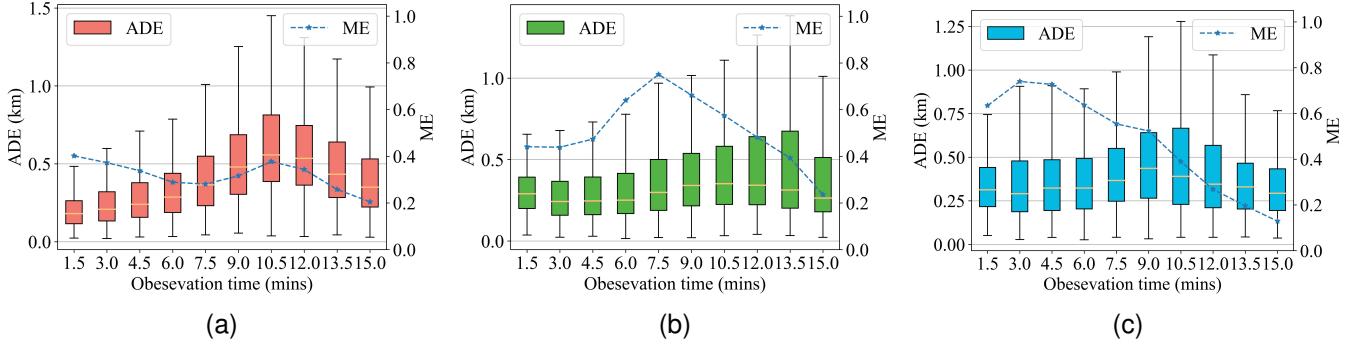


Fig. 9. Prediction error and model selection error for different observation length. (a) mode 1, (b) mode 2, (c) mode 3.

set of factors that are proportional to Gaussian distributions:

$$\begin{aligned} q(\mathbf{s}_{0:T}) &\propto \exp(\mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \boldsymbol{\Lambda}; \hat{\theta})]) \\ &\propto \mathcal{N}(\mathbf{s}_0 | \mathbf{m}_0, \mathbf{P}_0) \prod_{t=1}^T \underbrace{\mathcal{N}(\mathbf{s}_t | \mathbf{A}_t \mathbf{s}_{t-1}, (d\mathbf{V})^{-1})}_{q(\mathbf{s}_t | \mathbf{s}_{t-1})} \mathcal{N}(\mathbf{z}_t | \mathbf{B}\mathbf{s}_t, \mathbf{R}). \end{aligned} \quad (29)$$

In trajectory prediction, we are interested in the distributions of each individual state, i.e., the marginal distributions $q(\mathbf{s}_t)$. We obtain these using a filtering procedure, starting with $q(\mathbf{s}_0) = p(\mathbf{s}_0)$ and then recursively marginalizing out the previous state, i.e., $q(\mathbf{s}_t) = \int q(\mathbf{s}_t | \mathbf{s}_{t-1}) q(\mathbf{s}_{t-1}) d\mathbf{s}_{t-1}$. (15), 17 and 18 are based on the standard predict and update steps for Bayesian filtering with Gaussian distributions [42, Theorem 4.2], but use $(d\mathbf{V})^{-1}$ as process noise covariance matrix (as derived in (29) above).

In Equation (19), we claim the optimal variational distribution of the process noise precision matrix $\boldsymbol{\Lambda}$ is proportional to a

Wishart distribution. This derivation follows the same steps as those for the states. We start with the expectation over $q(\mathbf{s}_{0:T})$:

$$\begin{aligned} \mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{z}_{1:T}, \mathbf{s}_{0:T}, \boldsymbol{\Lambda}; \hat{\theta})] &= \mathbb{E}_{q_{\Lambda}}[\log p(\boldsymbol{\Lambda})] + \mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{s}_0)] \\ &+ \sum_{t=1}^T \mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{s}_t | \mathbf{s}_{t-1}, \boldsymbol{\Lambda}; \hat{\theta})] + \mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{z}_t | \mathbf{s}_t)]. \end{aligned} \quad (30)$$

Terms not involving $\boldsymbol{\Lambda}$, i.e., $\mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{s}_0)]$ and $\mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{z}_t | \mathbf{s}_t)]$, become part of the normalization constant and may be ignored for now. In the term involving the prior distribution, the expectation over the states drops out: $\mathbb{E}_{q_{\Lambda}}[\log p(\boldsymbol{\Lambda})] = \log p(\boldsymbol{\Lambda})$. The expectation over the state transition is:

$$\begin{aligned} \mathbb{E}_{q_{\Lambda}}[\log p(\mathbf{s}_t | \mathbf{s}_{t-1}, \boldsymbol{\Lambda}; \hat{\theta})] &= \\ \frac{1}{2} \log |\boldsymbol{\Lambda}| - \frac{1}{2} \mathbb{E}_{\mathbf{s}_t, \mathbf{s}_{t-1}}[(\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1})^T \boldsymbol{\Lambda} (\mathbf{s}_t - \mathbf{A}\mathbf{s}_{t-1})] + C. \end{aligned} \quad (31)$$

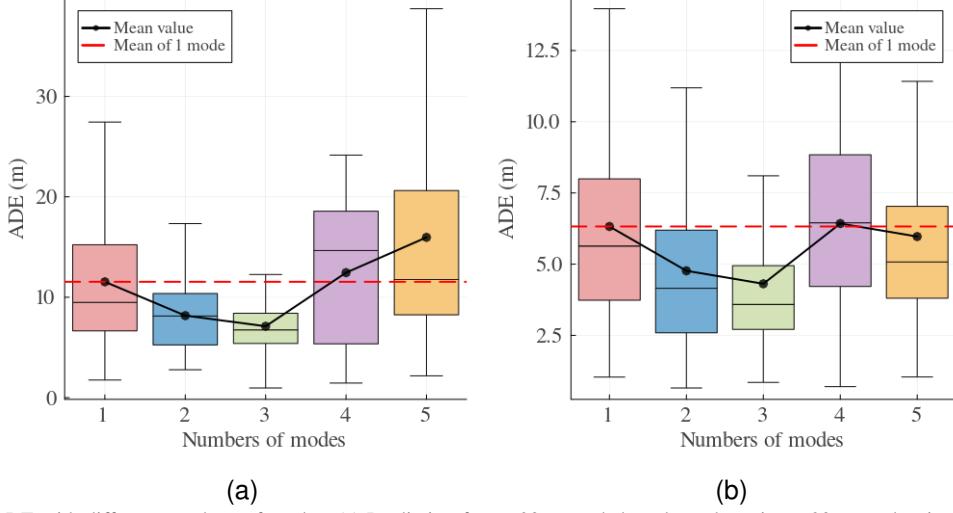


Fig. 10. Variation of ADE with different numbers of modes: (a) Predicting future 90 seconds based on observing a 90-second trajectory. (b) Predicting future 60 seconds based on observing a 120-second trajectory.

Completing the square and distributing the expectation leads to terms with products of states:

$$\mathbb{E}_{q_s}[\mathbf{s}_t^\top \Lambda \mathbf{s}_t] = \text{tr}(\Lambda(\mathbf{m}_t \mathbf{m}_t^\top + \mathbf{P}_t)) \quad (32)$$

$$\mathbb{E}_{q_s}[\mathbf{s}_t^\top \Lambda \mathbf{A} \mathbf{s}_{t-1}] = \text{tr}(\Lambda \mathbf{A} (\mathbf{m}_{t-1} \mathbf{m}_t + \mathbf{P}_{t-1})) \quad (33)$$

$$\mathbb{E}_{q_s}[\mathbf{s}_{t-1}^\top \mathbf{A}^\top \Lambda \mathbf{A} \mathbf{s}_{t-1}] = \text{tr}(\Lambda \mathbf{A} (\mathbf{m}_{t-1} \mathbf{m}_{t-1} + \mathbf{P}_{t-1}) \mathbf{A}^\top). \quad (34)$$

where $\text{tr}()$ represents the trace of the matrix. we have made use of the Gaussian joint distribution between two states

$$q(\mathbf{s}_{t-1}, \mathbf{s}_t) = \mathcal{N}\left(\begin{bmatrix} \mathbf{s}_{t-1} \\ \mathbf{s}_t \end{bmatrix} \mid \begin{bmatrix} \mathbf{m}_{t-1} \\ \mathbf{m}_t \end{bmatrix}, \begin{bmatrix} \mathbf{P}_{t-1} & \mathbf{P}_{t-1} \mathbf{A}^\top \\ \mathbf{A} \mathbf{P}_{t-1} & \mathbf{P}_t \end{bmatrix}\right). \quad (35)$$

Putting it all back together gives:

$$\begin{aligned} \mathbb{E}_{q_s}[\log p(\mathbf{s}_t \mid \mathbf{s}_{t-1}, \Lambda; \hat{\theta})] = & \\ \frac{1}{2} \log |\Lambda| - \frac{1}{2} (\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1})^\top \Lambda (\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1}) & \\ - \frac{1}{2} \text{tr}(\Lambda (\mathbf{P}_t - \mathbf{A} \mathbf{P}_{t-1} - \mathbf{P}_{t-1} \mathbf{A}^\top + \mathbf{P}_{t-1})) + C. & \end{aligned} \quad (36)$$

Exponentiating the above gives a density function that is proportional to a Wishart distribution, $\mathcal{W}(\Lambda \mid d_t, \mathbf{V}_t)$ where

$$d_t = l + 2 \quad (37)$$

$$\begin{aligned} \mathbf{V}_t = & (\mathbf{P}_t - \mathbf{A} \mathbf{P}_{t-1} - \mathbf{P}_{t-1} \mathbf{A}^\top + \mathbf{P}_{t-1}) \\ & + (\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1})(\mathbf{m}_t - \mathbf{A} \mathbf{m}_{t-1})^\top. \end{aligned} \quad (38)$$

where l is the dimensionality of the Wishart distribution. The product of two l -dimensional Wishart distributions is itself Wishart distributed:

$$\begin{aligned} \mathcal{W}(\Lambda \mid \mathbf{V}_1, d_1) \cdot \mathcal{W}(\Lambda \mid \mathbf{V}_2, d_2) & \\ \propto \mathcal{W}(\Lambda \mid (\mathbf{V}_1^{-1} + \mathbf{V}_2^{-1})^{-1}, d_1 + d_2 - l - 1). & \end{aligned} \quad (39)$$

In our case, the product of the Wishart prior and the Wishart distributed terms based on the state transitions in (30) is $\mathcal{W}(\Lambda \mid d, \mathbf{V})$ where

$$d = d_0 + \sum_{t=1}^T [d_t - l - 1] \quad (40)$$

$$\mathbf{V} = (\mathbf{V}_0^{-1} + \sum_{t=1}^T \mathbf{V}_t^{-1})^{-1}. \quad (41)$$

These are the parameters described after Equation (19).

REFERENCES

- [1] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett, “Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2313–2330, 2016.
- [2] G. Kavallieratos, V. Diamantopoulou, and S. K. Kat-sikas, “Shipping 4.0: Security requirements for the cyber-enabled ship,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6617–6625, 2020.
- [3] F. Castaldo, F. A. N. Palmieri, and C. S. Regazzoni, “Bayesian analysis of behaviors and interactions for situation awareness in transportation systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 313–322, 2016.
- [4] F. Castaldo and F. A. N. Palmieri, “Target tracking using factor graphs and multi-camera systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1950–1960, 2015.
- [5] J. Zhang, Y. Cui, G. Li, and J. Ren, “Dynamic path planning algorithm for unmanned surface vehicle under island-reef environment,” *IEEE Transactions on Aerospace and Electronic Systems*, 2023.

- [6] C. Veibäck, J. Olofsson, T. R. Lauknes, and G. Hen-deby, “Learning target dynamics while tracking using gaussian processes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 2591–2602, 2020. DOI: 10.1109/TAES.2019.2948699.
- [7] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, “Multimodal motion prediction with stacked transformers,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7577–7586.
- [8] J. Ma, C. Jia, Y. Shu, K. Liu, Y. Zhang, and Y. Hu, “Intent prediction of vessels in intersection waterway based on learning vessel motion patterns with early observations,” *Ocean Engineering*, vol. 232, p. 109 154, 2021.
- [9] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca, and P. Willett, “Deep learning methods for vessel trajectory prediction based on recurrent neural networks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4329–4346, 2021.
- [10] M. S. Wiig, K. Y. Pettersen, and T. R. Krogstad, “Collision avoidance for underactuated marine vehicles using the constant avoidance angle algorithm,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 951–966, 2019.
- [11] W. Zhang, Z.-J. Zou, and D.-H. Deng, “A study on prediction of ship maneuvering in regular waves,” *Ocean engineering*, vol. 137, pp. 367–381, 2017.
- [12] L. P. Perera, P. Oliveira, and C. G. Soares, “Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1188–1200, 2012.
- [13] W. Shaobo, Z. Yingjun, and L. Lianbo, “A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method,” *Ocean Engineering*, vol. 215, p. 107 910, 2020.
- [14] H. Rong, A. Teixeira, and C. G. Soares, “Ship trajectory uncertainty prediction based on a gaussian process model,” *Ocean Engineering*, vol. 182, pp. 499–511, 2019.
- [15] S.-k. Zhang, G.-y. Shi, Z.-j. Liu, Z.-w. Zhao, and Z.-l. Wu, “Data-driven based automatic maritime routing from massive ais trajectories in the face of disparity,” *Ocean Engineering*, vol. 155, pp. 240–250, 2018.
- [16] T. Wang, G. Li, L. I. Hatledal, R. Skulstad, V. Åesøy, and H. Zhang, “Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1781–1789, 2021.
- [17] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Prediction of vessel trajectories from ais data via sequence-to-sequence recurrent neural networks,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 8936–8940.
- [18] Y. Xiao, X. Li, W. Yao, J. Chen, and Y. Hu, “Bidirectional data-driven trajectory prediction for intelligent maritime traffic,” *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [19] L. You, S. Xiao, Q. Peng, et al., “St-seq2seq: A spatio-temporal feature-optimized seq2seq model for short-term vessel trajectory prediction,” *IEEE Access*, vol. 8, pp. 218 565–218 574, 2020.
- [20] B. Murray and L. P. Perera, “A dual linear autoencoder approach for vessel trajectory prediction using historical ais data,” *Ocean Engineering*, vol. 209, p. 107 478, 2020.
- [21] S. Capobianco, N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Recurrent encoder-decoder networks for vessel trajectory prediction with uncertainty estimation,” *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [22] N. Forti, L. M. Millefiori, P. Braca, and P. Willett, “Model-based deep learning for maneuvering target tracking,” in *2023 26th International Conference on Information Fusion (FUSION)*, 2023, pp. 1–6.
- [23] D.-w. Gao, Y.-s. Zhu, J.-f. Zhang, Y.-k. He, K. Yan, and B.-r. Yan, “A novel mp-lstm method for ship trajectory prediction based on ais data,” *Ocean Engineering*, vol. 228, p. 108 956, 2021.
- [24] P.-F. Xu, C.-B. Han, H.-X. Cheng, C. Cheng, and T. Ge, “A physics-informed neural network for the prediction of unmanned surface vehicle dynamics,” *Journal of Marine Science and Engineering*, vol. 10, no. 2, p. 148, 2022.
- [25] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar, “Kalmannet: Neural network aided Kalman filtering for partially known dynamics,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.
- [26] M. Kanazawa, T. Wang, R. Skulstad, G. Li, and H. Zhang, “Knowledge and data in cooperative modeling: Case studies on ship trajectory prediction,” *Ocean Engineering*, vol. 266, p. 112 998, 2022.
- [27] J. Chen, J. Ye, C. Zhuang, Q. Qin, and Y. Shu, “Liner shipping alliance management: Overview and future research directions,” *Ocean & Coastal Management*, vol. 219, p. 106 039, 2022.
- [28] S. Särkkä and J. Hartikainen, “Non-linear noise adaptive Kalman filtering via variational Bayes,” in *IEEE International Workshop on Machine Learning for Signal Processing*, 2013, pp. 1–6.
- [29] K. P. Murphy, *Probabilistic machine learning: an introduction*. MIT press, 2022.
- [30] İ. Şenöz, T. van de Laar, D. Bagaev, and B. de Vries, “Variational message passing and local constraint manipulation in factor graphs,” *Entropy*, vol. 23, no. 7, p. 807, 2021.
- [31] K. Friston, L. Da Costa, N. Sajid, et al., “The free energy principle made simpler but not too simple,” *Physics Reports*, vol. 1024, pp. 1–29, 2023.
- [32] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of*

- the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
- [33] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- [34] J. Ma, W. Li, C. Jia, C. Zhang, and Y. Zhang, “Risk prediction for ship encounter situation awareness using long short-term memory based deep learning on internship behaviors,” *Journal of Advanced transportation*, vol. 2020, pp. 1–15, 2020.
- [35] M. Kanazawa, L. I. Hatledal, G. Li, and H. Zhang, “Co-simulation-based pre-training of a ship trajectory predictor,” in *International Conference on Software Engineering and Formal Methods*, Springer, 2021, pp. 173–188.
- [36] T. Mitsuyuki, *Shipmmg.jl*, <https://github.com/ShipMMG/ShipMMG.jl>, Accessed: March 5, 2024.
- [37] H. Li, J. Liu, R. W. Liu, N. Xiong, K. Wu, and T.-h. Kim, “A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis,” *Sensors*, vol. 17, no. 8, p. 1792, 2017.
- [38] D. Bagaev, A. Podusenko, and B. De Vries, “Rxinfer: A julia package for reactive real-time bayesian inference,” *Journal of Open Source Software*, vol. 8, no. 84, p. 5161, 2023.
- [39] Y. Suo, W. Chen, C. Claramunt, and S. Yang, “A ship trajectory prediction framework based on a recurrent neural network,” *Sensors*, vol. 20, no. 18, p. 5133, 2020.
- [40] K. A. Sørensen, P. Heiselberg, and H. Heiselberg, “Probabilistic maritime trajectory prediction in complex scenarios using deep learning,” *Sensors*, vol. 22, no. 5, p. 2058, 2022.
- [41] J. Sekhon and C. Fleming, “A spatially and temporally attentive joint trajectory prediction framework for modeling vessel intent,” in *Learning for Dynamics and Control*, PMLR, 2020, pp. 318–327.
- [42] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge university press, 2013.



Jie Ma received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2010. From 2017 to 2018, he was a Visiting Scholar with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, USA. He is currently a Professor with the School of Navigation, Wuhan University of Technology, Wuhan. His researches include Intelligent navigation and spatiotemporal services, supported by the National Natural Science Foundation of China.



Wouter M. Kouw received a PhD degree in Computer Science from TU Delft, Netherlands, and has worked as a Niels Stensen Fellow at Copenhagen University, Denmark. He is currently an assistant professor working at TU Eindhoven, Netherlands, where he studies Bayesian inference algorithms for signal processing, control and mobile robotics.



Chengfeng Jia is currently a Ph.D. student in school of navigation, Wuhan university of technology and the guest Ph.D. student in Department of Electrical Engineering, Eindhoven University of Technology. His research interests focus on Bayesian machine learning and intelligent ship Navigation.