drones

MDPI

Article

Underwater DVL Optimization Network (UDON): A Learning-Based DVL Velocity Optimizing Method for Underwater Navigation

Feihu Zhang [1,*], Shaoping Zhao [1], Lu Li [2] and Chun Cao [1]

[1] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; zhaoshaoping@mail.nwpu.edu.cn (S.Z.); caochun@mail.nwpu.edu.cn (C.C.)

[2] Technical Center Department, Norinco Group Testing and Research Institute, Xi'an 710000, China; lilu1589143@163.com

[*] Correspondence: feihu.zhang@nwpu.edu.cn; Tel.: +86-029-88492611

Abstract: As the exploration of marine resources continues to deepen, the utilization of Autonomous Underwater Vehicles (AUVs) for conducting marine resource surveys and underwater environmental mapping has become a common practice. In order to successfully accomplish exploration missions, AUVs require high-precision underwater navigation information as support. A Strapdown Inertial Navigation System (SINS) can provide AUVs with accurate attitude and heading information, while a Doppler Velocity Log (DVL) is capable of measuring the velocity vector of the AUVs. Therefore, the integrated SINS/DVL navigation system can furnish the necessary navigational information required by an AUV. In response to the issue of DVL being susceptible to external environmental interference, leading to reduced measurement accuracy, this paper proposes an end-to-end deep-learning approach to enhance the accuracy of DVL velocity vector measurements. The utilization of the raw measurement data from an Inertial Measurement Unit (IMU), which includes gyroscopes and accelerometers, to assist the DVL in velocity vector estimation and to refine it towards the Global Positioning System (GPS) velocity vector, compensates for the external environmental interference affecting the DVL, therefore enhancing the navigation accuracy. To evaluate the proposed method, we conducted lake experiments using SINS and DVL equipment, from which the collected data were organized into a dataset for training and assessing the model. The research results show that the DVL vector predicted by our model can achieve a maximum improvement of 69.26% in terms of root mean square error and a maximum improvement of 78.62% in terms of relative trajectory error.

Keywords: AUV; SINS; DVL; deep learning; underwater navigation

Academic Editors: Daxiong Ji, Asiya Khan, Pablo Borja, Dena Bazazian, Mohd Hisham Bin Nordin and Andrey V. Savkin

Received: 8 November 2024
Revised: 10 January 2025
Accepted: 12 January 2025
Published: 15 January 2025

Citation: Zhang, F.; Zhao, S.; Li, L.; Cao, C. Underwater DVL Optimization Network (UDON): A Learning-Based DVL Velocity Optimizing Method for Underwater Navigation. *Drones* **2025**, *9*, 56. https://doi.org/10.3390/drones9010056

1. Introduction

AUVs have emerged as a research focus in the field of marine exploration in recent years [1,2], as they are widely applicable to various underwater needs, such as seabed mapping, archaeological surveys, environmental monitoring, underwater infrastructure inspections, and military reconnaissance [3]. The prerequisite for an AUV to perform tasks in unknown waters is its ability to obtain sufficiently accurate navigation information (attitude, heading, velocity, etc.) to support the AUV in completing the required actions [4,5]. In water, the Global Navigation Satellite System (GNSS) can provide most of the required navigation information. However, due to the rapid attenuation of radio signals in water, underwater positioning cannot be accomplished, and GNSS is unable to provide accurate navigation information for AUVs [6]. Underwater navigation methods for AUVs generally include geomagnetic field matching navigation, inertial navigation, acoustic navigation, etc. Geophysical field matching methods obtain navigation information by matching real-time measured physical feature data with pre-stored map data. For instance, a multibeam echo sounder (MBES) can be utilized to acquire bathymetric maps for navigation information. While MBES offer high precision, they are also costly and cumbersome, which limits their practical application in geophysical field matching methods.

A single navigation method alone is often insufficient to meet the precision requirements for AUV navigation, so the combination of SINS and DVL is a commonly used navigation method for AUVs. The combined navigation approach allows the two sensors to complement each other's strengths. It uses the error propagation variance of the SINS as the system state equation, the DVL velocity measurement as the velocity reference, and the difference in velocity between SINS and DVL as the observation value. Subsequently, the SINS error states are estimated and compensated through a Kalman filter, therefore enhancing the navigation accuracy. Despite this, the accuracy of the SINS/DVL combined navigation system is still influenced by the precision of the SINS and DVL sensors [7–10].

SINS utilizes an IMU to collect navigation information and computes the velocity of the AUV through integration, which is further used for dead reckoning. This can provide the navigation information required for an AUV [11]. The advantage of the IMU sensor lies in its ability to operate independently, as it neither emits signals nor relies on external units. It offers excellent anti-interference capabilities, allowing for the acquisition of high-precision, low-latency, and high-rate navigation information in a short period of time [12]. However, due to the nature of integration, the cumulative error in navigation information tends to increase as time progresses [13]. To enhance the navigation accuracy of the SINS, scholars have conducted a series of studies. Properly calibrating the SINS equipment can significantly improve the accuracy of the navigation information it provides. Calibration methods have rapidly gained widespread attention, such as using a high-precision three-axis turntable and the rotation of the system's three sensitive axes to achieve the calibration of error angles between the two systems [14]. The improved system-level fitting calibration method can address the severe coupling errors in the estimation of the scale factor error of the SINS using high-precision Ring Laser Gyros (RLGs) [15]. Additionally, a full-parameter calibration method for a six-axis tilt inertial measurement unit can reduce the difficulty of calibrating the tilt inertial assembly [16]. Moreover, a rapid analytical coarse calibration algorithm can eliminate calibration errors caused by IMU orientation errors [17] and can even achieve high-precision calibration of the SINS in polar regions [18]. When the SINS collects data, it inevitably becomes subject to the influence of the system noise inherent in the sensors. To address random noise, the Kalman filter can be used for estimation and noise reduction [19], or the Kalman filter can be applied to filter the high-frequency components of multi-resolution decomposed signals [20]. Additionally, wavelet domains can be utilized for denoising with correlated thresholding [21]. Furthermore, optimization techniques based on iterative rules can employ neural-fuzzy systems to predict SINS positioning errors [22].

Acoustic navigation is primarily implemented through a DVL, which is a navigation method based on the Doppler effect. It estimates the velocity vector of the AUV by calculating the Doppler frequency shifts of multiple beams reflected in different directions [23]. Due to its characteristic of not accumulating errors over time, the DVL can provide real-time high-precision velocity measurements of the AUV. However, the stability of these measurements is susceptible to interference from the external environment [24]. Most DVL algorithms assume that the attitude of the AUV remains consistent from the time of beam emission to the time of reception [25]. However, if the attitude changes during this period, it can lead to beam measurement errors, affecting the accuracy of DVL measurements [13]. The factors that affect the measurement accuracy of the DVL primarily include scale factor errors, misalignment angles, and random noise errors. To address scale factor errors, current methods include GNSS-assisted calibration based on velocity and position observations [23], as well as the use of a robust Invariant Extended Kalman Filter (IEKF). This approach combines the linear error propagation equation of the DVL calibration model based on the Lie group SO(3) with the theory of Statistical Similarity Measure (SSM), which can reduce the impact of outliers on calibration accuracy [10].

To address the random noise errors in the DVL, an improved robust Kalman Filter (KF) algorithm can be used to process the measurement noise variance of each DVL beam separately [26]. Additionally, a fault diagnosis method based on the chi-squared rule and a velocity tracking method based on a constant velocity model and the assumption of slow movement of the AUV can effectively suppress the random noise generated by DVL measurements [27]. When DVL measurement anomalies or beam loss occur, leading to a decrease in DVL measurement accuracy, AI and data-driven methods can be employed for predictive regression of the DVL [28–30]. To address installation misalignment angles, in the early stages, the least squares method could be used for estimation [31]. Currently, online calibration using Kalman filtering [32] or optimal estimation methods based on algorithms such as Particle Swarm Optimization (PSO) can also be employed to calibrate the DVL [33].

The accuracy of integrated navigation systems is still affected by the measurement precision of various sensors, including those from the SINS and the DVL. Inspired by Cohen [34], to enhance underwater navigation accuracy, this paper proposes an end-to-end deep-learning framework to improve the velocity vector of the DVL. Using the raw measurement data from IMU sensors, the method refines the DVL velocity readings. It compensates for external environmental interferences affecting the DVL, therefore enhancing navigation accuracy. The Figure 1 is the conceptual framework of this paper. The network utilizes raw IMU data to optimize the DVL velocity vector. During the training process, the output of the network and the actual GPS velocity vector are used to calculate the loss function. Finally, the optimized DVL velocity vector is integrated through the navigation system to obtain the position. The GPS described in this paper uses RTK differential positioning, which can achieve centimeter-level accuracy.

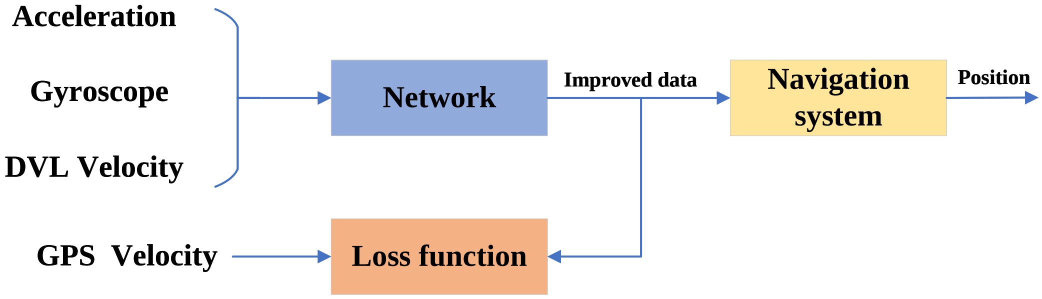


Figure 1. Overview of the proposed method.

To this end, we conducted experiments on a lake using both SINS and DVL equipment, collected data, and created a dataset to validate the effectiveness of the method proposed in this paper.

Summarizing the contributions of this paper as follows:

- We propose a deep-learning-based method for improving DVL measurements. This approach can compensate for external interferences during the DVL speed measurement process, therefore enhancing the accuracy of the DVL velocity vector.

- To facilitate further research in this field, we plan to make the test dataset and related code files publicly available and store them in a GitHub repository for everyone's use. (https://github.com/Graceful-ZSP/DVL-Improved.git) (accessed on 29 December 2024)

The remainder of this paper is organized as follows: Section 2 introduces the calculation principle of the DVL velocity vector and the model network framework proposed in this paper. Section 3 describes the relevant equipment used for data collection, the physical connections between the devices, and the location and process of data collection. Section 4 presents a qualitative and quantitative analysis of the results obtained from the model. Section 5 summarizes the work presented.

2. Principles and Network Framework

2.1. DVL Velocity Calculation

The DVL is an instrument that utilizes the Doppler effect of acoustic signals for velocity measurement and is widely used in the field of underwater navigation. The Doppler effect describes the phenomenon where there is a difference in frequency between the waves received and the waves emitted by a source when there is relative motion between the wave source and the receiver. The DVL emits multiple acoustic beams at various angles into the water and receives the echo signals that return after interacting with underwater objects, such as the sea floor. If there is relative motion, the frequency of the returning acoustic waves will change according to the Doppler effect. The DVL measures its velocity relative to the water column or the seafloor by analyzing the frequency difference (Doppler shift) between the emitted sound waves and the received echoes. It is represented by Equation (1):

$$f_r = f_t \left(\frac{1 \mp \frac{v_{beam}}{c}}{1 \pm \frac{v_{beam}}{c}} \right) \tag{1}$$

where f_r is the received frequency, f_t is the transmitted frequency, v_{beam} is the velocity of the beam relative to the medium, c is the speed of sound in the medium (water). Considering that the speed of the DVL is much slower than the speed of sound, the change in frequency can be simplified to Equation (2):

$$\Delta f \approx \frac{2 f_t v_{beam}}{c} \tag{2}$$

Each beam's velocity in a specific direction can be represented by Equation (3):

$$v_{beam} = \frac{c}{2 f_t} \Delta f \tag{3}$$

The sensor configuration of the DVL is in an X-shaped arrangement, also known as the Janus Doppler configuration. The Figure 2 shows a schematic diagram of the DVL beam directions:

By observing the geometric relationship between the DVL beams and the DVL body, the directions of the beams in the DVL body frame can be represented as (4):

$$b_t = \begin{bmatrix} \cos \varphi_i \sin \alpha & \sin \varphi_i \sin \alpha & \cos \alpha \end{bmatrix}_{1 \times 3} \tag{4}$$

where $(i = 1, 2, 3, 4)$ represents the beam number, and φ and α are the yaw and pitch angles relative to the vehicle body frame, respectively. All four beams have the same pitch angle, while the yaw angle is expressed as Equation (5):

$$\varphi_i = (i-1)\frac{\pi}{2} + \frac{\pi}{4} \tag{5}$$

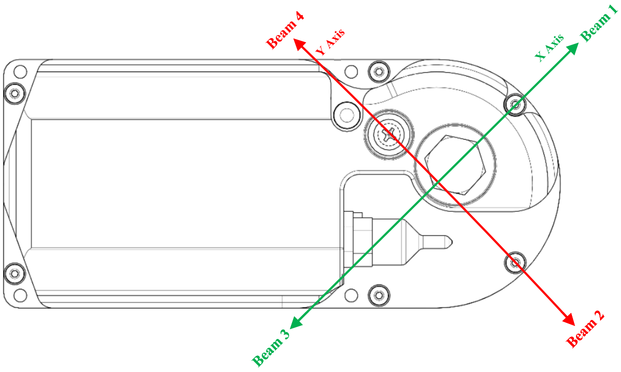


Figure 2. A schematic diagram of the DVL's four beams, which are arranged in an X-shaped layout.

We can define a transformation matrix B as shown in Equation (6):

$$B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}_{4\times 3} \tag{6}$$

Then, the relationship between the DVL velocity v_b^b and the beam velocity measurements v_{beam} can be represented through matrix B as shown in Equation (7):

$$v_{beam} = Bv_b^b \tag{7}$$

Based on the aforementioned relationship, using the least squares (LS) estimator, the DVL estimated velocity is obtained as shown in Equation (8):

$$\hat{v}_b^b = \underset{v_b^b}{argmin}\left\| y - Bv_b^b \right\|^2 \tag{8}$$

where y is the measured value of the beam velocity, based on the ideas of Braginsky and others [35], Equation (8) can be transformed into Equation (9).

$$\hat{v}_b^b = (B^T B)^{-1} B^T y \tag{9}$$

At this point, the three-axis velocity of the DVL carrier vehicle can be solved through the beam velocities measured by the DVL. Most DVL devices use Equation (9) to measure velocity. We aim to improve the velocity output of the DVL, hence it is necessary to select an error model. During the measurement process, the DVL may have scale factor errors, random noise, and misalignment angle errors. By incorporating these into Equation (7), the common error model of the DVL can be obtained [36], as shown in Equation (10):

$$y = B[v_b^b(1 + s_{DVL})] + b_{DVL} + n \tag{10}$$

where s_{DVL} is the scale factor, b_{DVL} is the bias vector, and n is zero-mean Gaussian white noise.

2.2. Deep-Learning Network Framework Based on IMU/DVL Integration

The internal IMU sensors of a SINS typically include three-axis gyroscopes and three-axis accelerometers. The raw measurement data from these gyroscopes and accelerometers can be used to calculate the position, orientation, and velocity of the AUV in space, therefore capturing the dynamic information of the AUV's motion.

Liu and others [37] proposed a deep-learning network architecture known as the GPS/SINS neural network, which combines Convolutional Neural Networks (CNN) and Gated Recurrent Unit (GRU) neural networks. This architecture extracts spatial features from IMU signals and tracks their temporal characteristics, therefore enhancing navigation performance during GPS outages. This demonstrates that IMU sensors contain sufficient navigation information, which is why this paper utilizes the output data from gyroscopes and accelerometers to assist in improving the DVL's velocity vector.

GPS provides extremely high accuracy in position and velocity, and in many studies, the position and velocity information from GPS are considered a high-precision reference standard. Therefore, in this paper, the GPS velocity output is used as the target for improving the DVL output velocity. By continuously refining the calibration of DVL measurements, the reliability of the DVL output velocity is enhanced. Radio waves suffer significant attenuation in water, which impedes the underwater transmission path of GPS, leading to signal transmission delays or even loss. Therefore, we opt to enhance the forward and lateral velocities of the DVL, using the forward and lateral velocities from GPS as the targets for improvement. Data collection on the lake is simplified because it only requires a test vessel to sail on the surface to collect the required data.

The output rate of the measurement data from the gyroscope and accelerometer is 100 Hz, while the output rate of the measurement data from the DVL sensor is only 5 Hz. This means that we cannot directly input the data from the three sensors into the network. Therefore, following the approach in Cohen [34], we package the data from the gyroscope and accelerometer according to the ratio of the output rates of the three sensors, which is 20:20:1. That is to say, every 20 sets of measurements from the gyroscope and accelerometer correspond to one set of measurements from the DVL.

We first pass the raw measurements from the gyroscope and accelerometer through a one-dimensional convolutional layer (1D Conv) composed of six 2 × 1 filters to effectively extract spatial features from the sequential data and generate feature vectors. The selection of a kernel length of 2 is a comprehensive consideration of computational efficiency and feature capture capabilities, while the design of 6 kernels is to achieve multi-scale feature fusion. Then, a Flatten layer is applied to "flatten" the multi-dimensional feature vectors into one-dimensional data. The output of the one-dimensional data is then passed through a Long Short-Term Memory (LSTM) layer. The LSTM layer can capture temporal features of the data, allowing for further feature extraction based on the temporal dynamics. The two sets of feature data are then combined and passed through a Dropout layer, which randomly discards a certain proportion of the data to reduce the risk of model overfitting. Afterward, the data goes through a series of fully connected layers. The output is combined with the current DVL velocity vector and, through the final fully connected layer, produces a 2 × 1 vector. This vector represents the predicted DVL velocity vector by the model. The specific architecture of the model is shown in Figure 3 below.

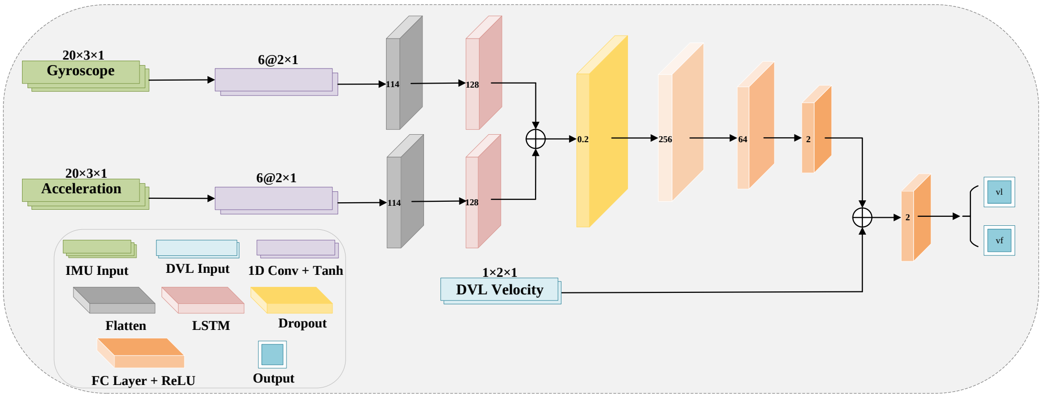


Figure 3. Deep-learning network architecture diagram of the proposed model in this paper.

For the aforementioned network architecture, the following definitions are utilized:

- Flatten Layer: Transforms multi-dimensional input data into a one-dimensional format.
- Dropout Layer: Regularizes the input data to reduce the risk of model overfitting.
- Linear Layer: Performs a linear transformation on the input data, integrating features and producing an output of a fixed size.
- Tanh Activation Function: The hyperbolic tangent function maps the input value x to a nonlinear range between -1 and 1. It is specifically implemented as:

$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{11}$$

- ReLU (Rectified Linear Unit) Activation Function: Also known as the ramp function, it is a nonlinear function with both biological and mathematical foundations, and it is specifically implemented as:

$$\mathrm{ReLU}(x) = max(0, x) \tag{12}$$

- 1D Conv Layer: Creates a convolutional kernel that captures local features of the input data along a single spatial dimension through convolution and outputs a feature vector. In a 1D Conv layer, all input parameters interact directly with the output. The relationship between the input and output is as follows:

$$y_t = \sum_{p}^{i=1} x_{t+i} + w_i \tag{13}$$

  where x is input, y is output, t represents the timestamp, p denotes the kernel length. The p is the size of the filter or kernel used in the convolutional operation. The w symbolizes the learned weights.
- LSTM Layer: The LSTM layer can learn the long-term dependencies in the input data and extract the temporal features of the data through its internal gating structure. It is an advanced version of the Recurrent Neural Network (RNN) because it solves the problem of gradient explosion in time series [38]. The LSTM uses input gates, output gates, and forget gates to extract temporal features. The role of the forget gate is to discard irrelevant information from previous outputs and current inputs, represented as:

$$f_t = \sigma(x_t U^f + h_{t-1} W^f + b_f) \tag{14}$$

where x_t is the current input, h_{t-1} is the output from the previous moment, W^f and U^f are the forget gate weights, and b_f is the bias. The role of the input gate is to update the cell state, represented as:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i + b_i) \tag{15}$$

where W^i and U^i are the input gate weights, and b_i is the bias. The role of the output gate is to determine the output and decide which information from the cell state to pass on, represented as:

$$o_t = \sigma(x_t U^o + h_{t-1} W^i + b_o) \tag{16}$$

where W^o and U^o are the input gate weights, and b_o is the bias. The LSTM layer outputs the current output and hidden state, represented as the result of the output gate multiplied by the cell state through a tanh layer:

$$h_t = o_t \cdot Tanh(C_t) \tag{17}$$

where C_t is the current cell state, represented as Equation (18).

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \tag{18}$$

where $\tilde{C}_t$ is the estimated cell state, represented as Equation (19)

$$\tilde{C}_t = Tanh(x_t U^g + h_{t-1} W^g + b_g) \tag{19}$$

where W^g and U^g are the gate weights, and b^g is the bias.

The CNN-LSTM is a hybrid neural network architecture that combines the strengths of both CNN and LSTM. This architecture leverages the CNN's proficiency in handling spatial data, such as images, and the LSTM's expertise in processing sequential data, such as time series. It is widely used in fields like image captioning, video analysis, and time series forecasting. CNN excels at extracting spatial features, while LSTMs are adept at handling sequential features. By inputting the spatial features extracted by the CNN into the LSTM, it is possible to model and analyze sequences of spatial features. The CNN-LSTM architecture, by combining the advantages of both CNN and LSTM, exhibits stronger generalization capabilities and expressiveness when dealing with complex spatiotemporal data. The IMU, DVL, and GPS data we collected are all time series data with inherent spatial characteristics. In response to this scenario, we chose the CNN-LSTM architecture to improve the DVL velocity vector.

2.3. Network Hyperparameter Definition

Mean Squared Error (MSE) measures the accuracy of model predictions by calculating the average of the squares of the differences between the predicted values and the actual values. It is sensitive to the values of the Lie group and has high computational efficiency, which is suitable for the scenario of underwater navigation, so we use it as the model's loss function. The definition of MSE is shown in Equation (20) as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i+1}^{n} (y_{actual} - y_{predicted})^2 \tag{20}$$

where n is the number of samples, y_{actual} is the target value, and $y_{predicted}$ is the model's predicted value. The forward propagation process during training is the process by which input data passes through all layers of the network architecture Figure 3. After completing the forward propagation, the loss function is calculated, and then the weights and biases of all layers are updated during the backpropagation process using the gradient descent method. The principle of gradient descent is defined as:

$$\theta = \theta - \eta \nabla_{\theta} J(\theta) \tag{21}$$

where θ is the vector of weights and biases, $J(\theta)$ is the value of the loss function computed by the model when the weights and biases are set to θ, ∇_{θ} is the gradient operator, and η is the learning rate.

The learning rate holds an important position among all hyperparameters, as it directly affects the speed of the model training process and the performance of the model. If the learning rate is too low, it will cause the model to fall into local optima too quickly. If it is too high, it will prevent the model from converging. We used the Adam optimizer to achieve adaptive adjustment of the learning rate, aiming to improve the efficiency and stability of model training. Specifically, we set an initial learning rate of 0.001 and allowed the Adam optimizer to adjust it automatically. Additionally, we selected MSE as the loss function and found in our experiments that the model typically converges after 500 iterations. Regarding the batch size, we chose 4 as a compromise to balance memory usage and model performance. The selection of these hyperparameters was based on the results of multiple experiments. To mitigate overfitting, we employed regularization techniques during model training. Specifically, we added an L2 regularization term to the loss function to constrain the model weights and prevent the model from becoming too complex.

3. Data Acquisition System Composition

3.1. Hardware Equipment

We opt to use a SINS to collect the required data from gyroscopes, accelerometers, and GPS. The SINS is equipped with high-precision fiber optic gyros, high-precision quartz flex accelerometers, and a mobile mapping-grade multi-mode and multi-frequency GNSS receiver supporting independent BeiDou functions, enabling high-precision measurement of the carrier's heading, attitude, velocity, and position. The DVL used for data collection is referred to as "Pathfinder". This model of DVL features a unique phased array sensor and does not require sound velocity correction. It is compact but can provide highly reliable velocity data. A field photo of the SINS and DVL devices is hlshown in Figure 4.

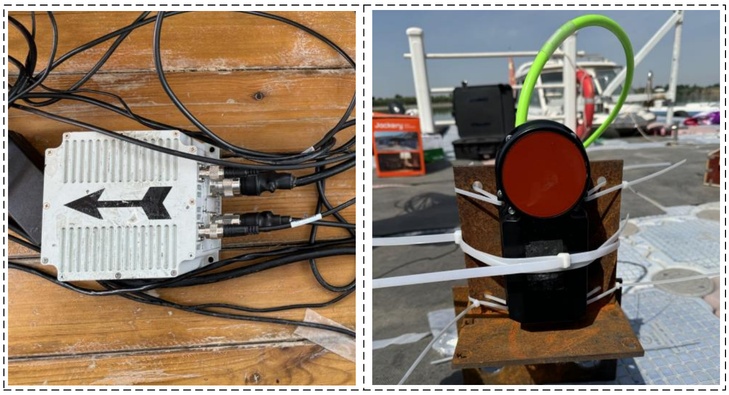


Figure 4. The device on the left is the SINS, and the device on the right is the DVL device.

Key performance parameters of the two devices are shown in the Table 1 below:

Table 1. Some key parameters of SINS and DVL devices.

Device	Parameter	Performance	Coordinate System
SINS	Heading accuracy Gyro zero deviation stability Accelerometer zero offset stability	0.05° $\leq$0.1°/h $\leq$20 μg	Forward, left, up
DVL	Frequency Velocity accuracy Altitude	614.4 kHz ±0.2% ±0.2 cm/s 0.2 m–89 m	Forward, left, up

3.2. Data Collection Methods

To verify the proposed method, a canyon located in the northwest of China to complete the data collection, meeting the needs of our experiment. We use the experimental vessel to complete 15 tasks in the canyon, with varying navigation trajectories and times for each task, including C-shaped, Mountain-shaped, round, M-shaped, N-shaped, wavy trajectories, etc. In the data collection process, we carefully selected a variety of shapes, including the M shape, to comprehensively capture different dynamic characteristics. The design of the M shape is particularly used to capture the dynamic parts of rapid changes and turns, which is crucial for evaluating the system's performance under complex motion patterns. In addition, other shapes, such as straight lines, curves, etc., are also included to cover motion situations at different speeds and accelerations. Through such diverse shape selection, we aim to ensure the richness and representativeness of the dataset, therefore improving the generalization ability of the model in different scenarios. We selected 4 tasks from a total of 15 tasks to form our test dataset, 1 task for model validation, and the remaining 10 tasks for training our model. The satellite image of the experimental area and the trajectory maps of the four tasks used as test data are shown in Figure 5.

The directions of the two devices are not consistent. We can align the directions of the two devices by self-compensating 45° through internal DVL commands. To ensure the mechanical stability of these two devices, we customized a mechanical structure for fixing the DVL on the test vessel. In the design of the experimental setup and the data processing, we have thoroughly considered and compensated for the leverage arm effect between sensors, ensuring the consistency and accuracy of the data. The schematic diagrams of our test vessel, equipment coordinate directions, and data collection process can be seen in Figure 6.

The dataset has a total duration of over 2 h, including 21,615 DVL measurement data points, 432,300 gyroscope and accelerometer measurement data points, and 432,300 GPS velocity measurement data points. The data volume for each task varies, with the data used for training to account for approximately 70% of the total dataset and the data used for testing making up about 20% of the total dataset. The validation dataset comprises approximately 10% of the total dataset.

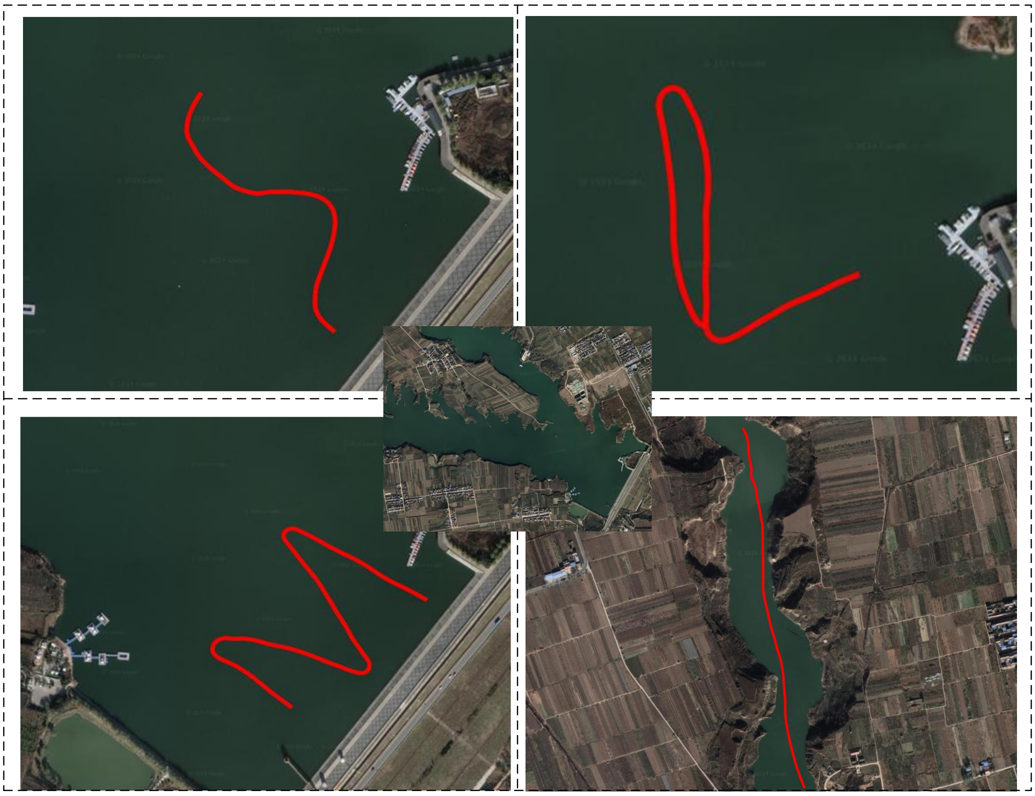


Figure 5. In the **top left** corner is the satellite trajectory map for Task-1, in the **top right** corner is Task-2, in the **bottom left** corner is Task-3, and in the **bottom right** corner is Task-4. The small map in the center is the satellite image of the experimental area.

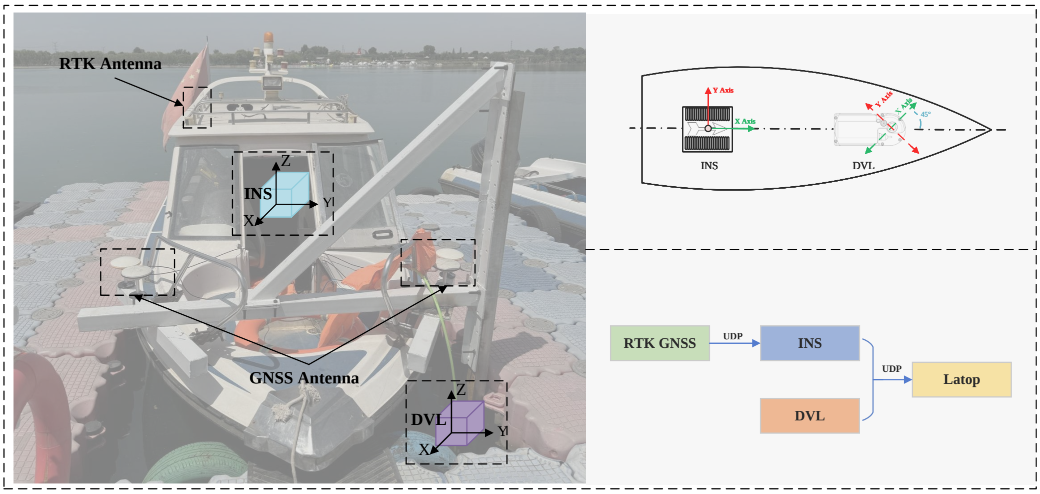
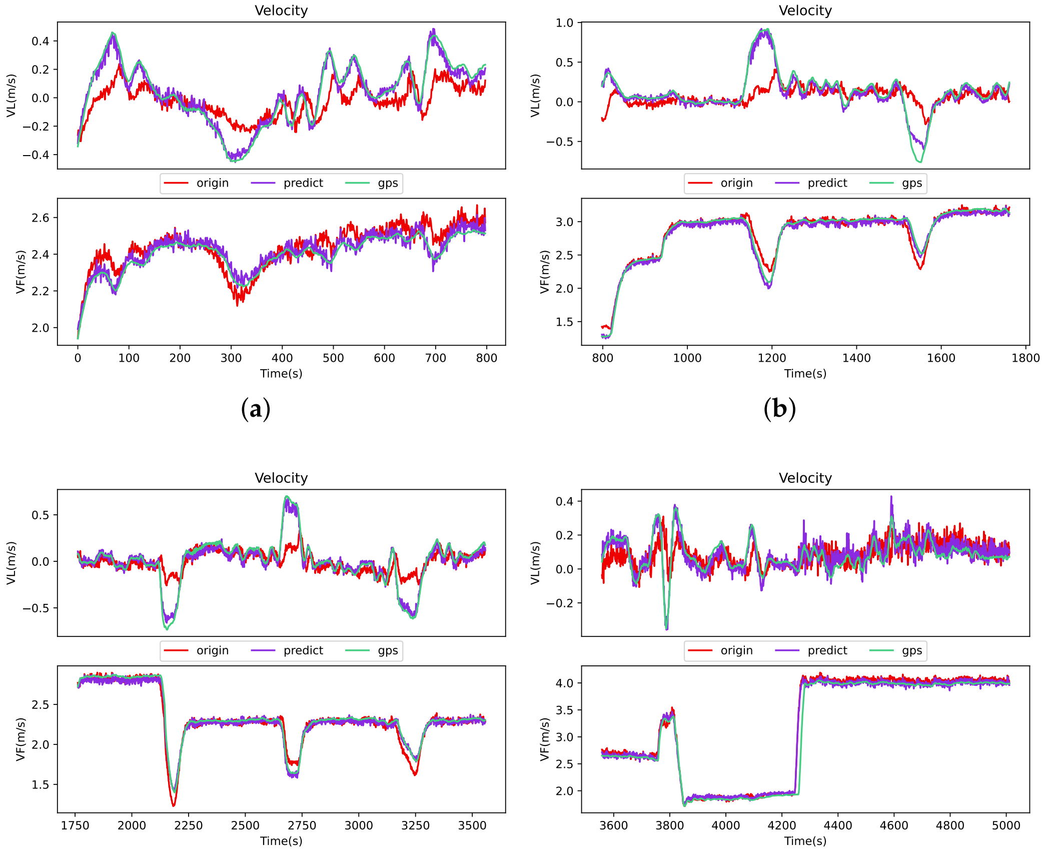


Figure 6. In the **left** panel is our test vessel with the fixed mechanical structure installed. In the **upper right** corner is the coordinate direction diagram for the SINS and DVL devices, and in the **lower right** corner is the schematic diagram of the data collection process.

4. Data Analysis

Underwater navigation systems utilize technologies such as acoustics, inertial measurement, and other sensing techniques to achieve positioning and navigation. The primary methods include acoustic navigation, inertial navigation, and integrated navigation. Acoustic navigation works on the principle of using the propagation characteristics of sound waves in water for positioning. Sound wave signals are emitted from a transmitter, reflect off obstacles, and by measuring the time and direction of the sound wave's propagation, the position of the object is determined. Inertial navigation calculates position and heading by measuring the device's acceleration and angular velocity. Inertial navigation systems do not rely on external signals, making them particularly useful in underwater environments. Integrated navigation combines various navigation technologies (such as GPS, DVL, acoustic navigation, etc.) to enhance navigation accuracy and reliability. For example, an INS/DVL/GPS-based integrated navigation system corrects the errors of the inertial navigation system by fusing data from multiple sources, therefore improving navigation precision.

The training set is used to train the model, with the batch size set to 4 and the learning rate set to n = 0.001. The learning rate is decayed by a factor of 0.5 every 150 training epochs, and the model converges after 500 training epochs. The test set is input into our well-trained model to obtain the predicted DVL velocity vector and to calculate the corresponding evaluation metrics. The performance of the test set on the model can be seen in Figure 7.



(**a**) (**b**)

(**c**) (**d**)

Figure 7. Comparison chart of the original DVL velocity, model-improved velocity, and GPS output velocity, where (**a**) Task-1; (**b**) Task-2; (**c**) Task-3; (**d**) Task-4. The VL represents the lateral velocity of DVL, and the VF represents the forward velocity of DVL.

It is evident from the velocity curves that the DVL velocity vectors predicted by the model closely follow the GPS velocity vectors, especially during significant changes in speed. In Figure 7, it is easy to see that the performance of the original DVL's VL is relatively poor, especially when changing direction. This is consistent with the speed measurement rules of the DVL and is the expected result, as the likelihood of the DVL's beam loss increases when there is a significant change in the direction of the vehicle, which can lead to poor quality of the speed vector output by the DVL. However, after optimization with our algorithm, we can better correct the DVL. The original DVL's VF performance is better, which may be because the direction of the vehicle is always forward. The quality decreases when there is a sudden change in VF, but after optimization with our algorithm, it can show a good state. In summary, our algorithm can significantly optimize the DVL speed vector.

The performance evaluation of underwater navigation systems is a crucial step in ensuring their reliable operation in complex marine environments. The most critical aspects of navigation are position and velocity, so evaluating the performance of an underwater navigation system essentially requires assessing the accuracy of its position and velocity measurements. The definition of the model evaluation metrics follows Armaghani and Asteris [39], and there are four in total:

1. Root Mean Squared Error (RMSE), a metric used to measure the difference between the model's predictions and the actual values. The smaller the metric, the smaller the difference between the model's predictions and the actual values, indicating better model prediction performance:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{22}$$

where x_i refers to the true value at time i, and $\hat{x}_i$ refers to the predicted value at time i.

2. Mean Absolute Error (MAE), a metric used to measure the prediction error of the model. The smaller the metric, the smaller the prediction error, indicating better model prediction performance:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\|x_i - \hat{x}_i\| \tag{23}$$

3. Coefficient of Determination (R^2), which indicates the strength of the relationship between two quantities. The closer the metric is to 1, the stronger the relationship between the two quantities, and the closer to 0, the weaker the relationship:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x}_i)^2} \tag{24}$$

where $\bar{x}_i$ represents the mean of the true values.

4. Variance Accounted For (VAF), a metric used to evaluate the predictive capability of the model. The larger the metric, the better the model's predictive ability:

$$\mathrm{VAF} = \left[1 - \frac{var(x_i - \hat{x}_i)}{var(x)_i}\right] \times 100 \tag{25}$$

By inputting the test set into the model, we calculated the four evaluation metrics for both the DVL original velocity vector and the model-predicted velocity vector relative to the GPS velocity vector. The specific results are shown in the Table 2 below:

Table 2. The performance of the DVL velocity vector and the improved network velocity vector on the four evaluation metrics, as well as the improvement ratio of the network. Task-1 is mountain-shaped, Task-2 is round, Task-3 is M-shaped, and Task-4 is wavy.

Indicators	Code Name	DVL	Proposed	Improvement
RMSE	Task-1	0.1132	**0.0348**	**69.26%**
	Task-2	0.1596	**0.0524**	67.17%
	Task-3	0.1242	**0.0390**	68.60%
	Task-4	0.1271	**0.0974**	23.37%

Table 2. *Cont.*

Indicators	Code Name	DVL	Proposed	Improvement
MAE	Task-1	0.0919	**0.0284**	**69.10%**
	Task-2	0.0959	**0.0396**	58.71%
	Task-3	0.0763	**0.0308**	59.63%
	Task-4	0.0663	**0.0432**	34.84%
R^2	Task-1	0.5077	**0.9341**	**83.99%**
	Task-2	0.5751	**0.9704**	68.74%
	Task-3	0.7091	**0.9792**	38.10%
	Task-4	0.4993	**0.8857**	77.39%
VAF	Task-1	62.6726	**95.1262**	51.78%
	Task-2	59.6023	**97.6146**	63.78%
	Task-3	71.2616	**98.2074**	37.81%
	Task-4	50.1909	**88.8289**	**76.98%**

From an objective evaluation of the velocity metrics, the model we proposed can significantly improve the DVL velocity vector. The RMSE achieved a maximum improvement of 69.26%, the MAE achieved a maximum improvement of 69.10%, the R^2 achieved a maximum improvement of 83.99%, and the VAF achieved a maximum improvement of 76.98%.

To further demonstrate the improvement in navigation accuracy, we performed dead reckoning on the carrier using the GPS velocity vector, the original DVL velocity vector, and the improved DVL velocity vector, and the resulting trajectory is shown in Figure 8 below.

In Figure 8, it is easy to see that the performance of the original SINS is poor. The reason for the poor performance of the SINS is also within our expectations, as the error of the SINS increases with the duration of navigation, so DVL is generally used to correct the velocity of the SINS. Although the DVL has a small error compared to GPS, there is still a certain error, especially after multiple changes in direction. The error increases, which also corresponds to the phenomenon in Figure 7 where the DVL's speed measurement is poor after changing direction. The DVL improved by our model is closer to GPS and has higher accuracy.

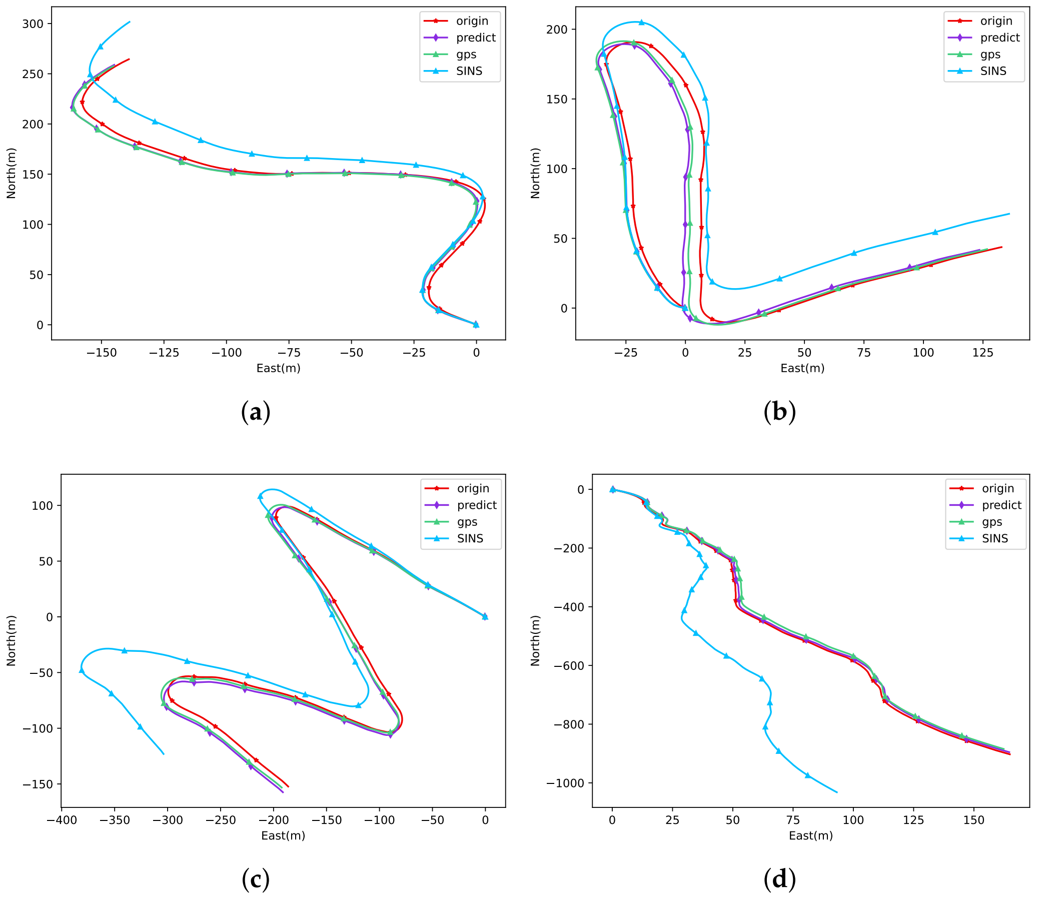
We use two metrics, Absolute Trajectory Error (ATE) and Relative Trajectory Error (RTE), to assess the differences between trajectories. ATE measures the overall difference of the trajectory estimation, focusing on the global discrepancy between the estimated trajectory and the true trajectory; RTE measures the local difference in trajectory estimation, focusing on the accuracy of adjacent position estimations within the trajectory. The expression for ATE is shown in Equation (26).

$$\mathrm{ATE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{26}$$

where x_i refers to the true value at time i, and $\hat{x}_i$ refers to the predicted value at time i. The expression for RTE is shown in Equation (27):

$$\mathrm{RTE} = \frac{1}{n}\sqrt{\sum_{i=1}^{n-1}[(x_i - x_{i+1}) - (\hat{x}_i - \hat{x}_{i+1})]^2} \tag{27}$$

where x_{i+1} refers to the true value at time $i+1$, and $\hat{x}_{i+1}$ refers to the predicted value at time $i+1$. Based on the trajectory data, we can calculate the ATE and RTE of the DVL original velocity dead-reckoned trajectory and the model-predicted dead-reckoned trajectory with respect to the GPS velocity dead-reckoned trajectory. These results are shown in Table 3.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 8. The trajectory obtained by dead reckoning using the original DVL velocity, original SINS velocity, model-improved velocity, and GPS output velocity, where (**a**) Task-1; (**b**) Task-2; (**c**) Task-3; (**d**) Task-4. East represents the displacement of the vehicle in the eastward direction, and North represents the displacement of the vehicle in the northward direction.

From the objective point of view of ATE and RTE indicators, the improved DVL speed vector of the model has a good enhancement effect on the navigation accuracy, with a maximum increase of 78.62% in ATE and 69.90% in RTE.

In summary, the original DVL velocity vector, after being improved by the model we proposed, can become closer to the GPS velocity vector, therefore enhancing the accuracy of underwater navigation.

Table 3. The performance of the DVL velocity vector, the improved network velocity vector, and the SINS inherent velocity vector in dead reckoning on two evaluation metrics.

Indicators	Code Name	DVL	Proposed	Improvement	SINS
ATE	Task-1	4.0303	**0.8616**	**78.62%**	141.5338
	Task-2	5.3697	**1.7438**	67.53%	88.3024
	Task-3	4.3559	**3.0779**	29.34%	168.9872
	Task-4	9.2006	**6.3981**	30.46%	371.1152
RTE	Task-1	0.0289	**0.0087**	**69.90%**	0.0588
	Task-2	0.0309	**0.0126**	59.22%	0.0345
	Task-3	0.0243	**0.0098**	59.67%	0.0730
	Task-4	0.0215	**0.0142**	33.95%	0.1126

5. Conclusions and Prospects

5.1. Conclusions

We proposed a deep-learning method assisted by an IMU to enhance DVL velocity vectors. We utilize raw data from IMU sensors (gyroscopes and accelerometers) to assist in refining the original DVL velocity vectors, aiming to improve the accuracy of underwater navigation. To evaluate the effectiveness of this method, we conducted lake experiments using related equipment such as SINS and DVL, completing multiple tasks with varying navigation trajectories and times, and all task data collected served as a dataset for model training and testing. The results of the model on the test set show significant improvements in the four evaluation metrics of RMSE, MAE, R^2, and VAF for the DVL velocity vector improved by the model. On the velocity map, it can be seen that the original DVL velocity vector does not perform well when the carrier turns, and the model's improved DVL velocity vector handles this situation well. On the dead-reckoning trajectory map, it is evident that the trajectory of the carrier derived from the DVL velocity vector predicted by the model has a higher degree of overlap with the trajectory derived from the GPS velocity vector, and the ATE and RTE between the two trajectories are smaller.

5.2. Prospects

This paper only considered the state during constant-depth navigation, so we have only improved the DVL's lateral and forward velocity vectors. In the future, we plan to incorporate depth data into the model, therefore improving the DVL's three-axis velocity and further enhancing the accuracy of underwater navigation.

The conclusions of this paper are based on lake environments. It is well known that there are significant differences between lake environments and marine environments. Both acidity and temperature can affect the speed measurement of DVL, and marine environments tend to have larger waves. The applicability of the method proposed in this paper to marine environments remains to be considered. In the future, we will actively make improvements when conditions allow and complete the validation in marine environments.

The experimental analysis results presented in this paper demonstrate the effectiveness of the proposed method but do not elucidate whether it has an advantage over other similar methods. Additionally, this paper does not provide evidence to indicate that machine learning methods may incur relatively higher computational costs compared to more traditional numerical processing methods, such as filtering. Subsequent researchers can continue to investigate in this area.

This paper only considered the fixed SINS and DVL, lacking consideration for the mobility of the equipment. In the future, we will also focus on the transferability of the equipment, mainly considering the following three points:

1. Transferability between different DVL: Different DVL products are bound to affect the transferability of the model. In the future, the robustness of the model can be enhanced to achieve compatibility with different models of DVL products.
2. Transferability between different installation settings: When installing DVL and SINS, it is necessary to pay attention to the compensation of the lever arm and perform navigation after the SINS alignment, which can improve the stability of the equipment.
3. Transferability between SINS changes: Different SINS will also affect performance, but the better the performance of the SINS, the greater the improvement effect on the DVL. In the future, the performance of the model can be indirectly improved by improving the performance of the SINS.

We can also improve the transferability of the algorithm through certain measures, which may include developing more general-purpose algorithms, conducting more cross-platform tests, and establishing more comprehensive error compensation models.

Author Contributions: Conceptualization, S.Z., C.C. and F.Z.; methodology, S.Z. and C.C.; validation, S.Z.; data curation, S.Z. and C.C.; writing—original draft preparation, S.Z.; writing—review and editing, S.Z., C.C., F.Z. and L.L.; visualization, S.Z., L.L. and C.C.; supervision, F.Z.; project administration, F.Z. and L.L.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (NSFC), grant number 52171322.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors gratefully acknowledge the support provided by the Key Laboratory of Unmanned Underwater Transport Technology during the data collection process and the assistance of the research team members.

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

Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
SINS	Strapdown Inertial Navigation System
DVL	Doppler Velocity Log
GPS	Global Positioning System
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GNSS	Global Navigation Satellite System
MBES	Multibeam echo sounder
KF	Kalman Filter
AI	artificial intelligence
LS	Least Squares
GRU	Gated Recurrent Unit
1D Conv	one-dimensional convolutional
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
R^2	Coefficient of Determination
VAF	Variance Accounted For
ATE	Absolute Trajectory Error
RTE	Relative Trajectory Error

References

1. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: A review. *IEEE J. Ocean. Eng.* **2013**, *39*, 131–149. [CrossRef]
2. Wynn, R.; Huvenne, V.; Le Bas, T.; Murton, B.J.; Connelly, D.P.; Bett, B.J.; Ruhl, H.A.; Morris, K.J.; Sumner, E.J.; Darby, S.E.; et al. Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Mar. G Eology* **2014**, *352*, 451–468. [CrossRef]
3. Zhang, B.; Ji, D.; Liu, S.; Zhu, X.; Xu, W. Autonomous underwater vehicle navigation: A review. *Ocean Eng.* **2023**, *273*, 113861. [CrossRef]
4. Mu, X.; He, B.; Wu, S.; Zhang, X.; Song, Y.; Yan, T. A practical INS/GPS/DVL/PS integrated navigation algorithm and its application on Autonomous Underwater Vehicle. *Appl. Ocean Res.* **2021**, *106*, 102441. [CrossRef]

5. Ding, S.; Ma, T.; Li, Y.; Xu, S.; Yang, Z. RD-VIO: Relative-depth-aided visual-inertial odometry for autonomous underwater vehicles. *Appl. Ocean Res.* **2023**, *134*, 103532. [CrossRef]
6. Miller, A.; Miller, B.; Miller, G. AUV position estimation via acoustic seabed profile measurements. In Proceedings of the 2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV), Porto, Portugal, 6–9 November 2018.
7. Luo, L.; Zhang, Y.; Fang, T.; Li, N. A new robust Kalman filter for SINS/DVL integrated navigation system. *IEEE Access* **2019**, *7*, 51386–51395. [CrossRef]
8. Wang, D.; Xu, X.; Huo, L. An improved adaptive Kalman filter for underwater SINS/DVL system. *Math. Probl. Eng.* **2020**, *2020*, 5456961. [CrossRef]
9. Liu, R.; Liu, F.; Liu, C.; Zhang, P. Modified Sage-Husa Adaptive Kalman Filter-Based SINS/DVL Integrated Navigation System for AUV. *J. Sens.* **2021**, *2021*, 9992041. [CrossRef]
10. Xu , B.; Guo, Y.; Hu, J.; Zhang, P. An improved robust Kalman filter for SINS/DVL tightly integrated navigation system. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 8502915. [CrossRef]
11. Dang, L.; Chen, B.; Huang, Y.; Zhang, Y.; Zhao, H. Cubature Kalman filter under minimum error entropy with fiducial points for INS/GPS Integration. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 450–465. [CrossRef]
12. Melo, J.; Matos, A.; Liu, S.; Zhu, X. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Eng.* **2017**, *139*, 250–264. [CrossRef]
13. Miller, P.; Farrell, J.; Zhao, Y.; Djapic, V. Autonomous underwater vehicle navigation. *IEEE J. Ocean. Eng.* **2010**, *35*, 663–678. [CrossRef]
14. Gao, X.; Lv, Y. The technique of installation error angle calibration in the experiment of accuracy evaluation in INS. In Proceedings of the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 3–5 October 2017.
15. Liu, X.; Zheng, J.; Lu, J.; Zhao, V.; Ye, L. Reducing the Effect of the Accelerometer-Slope Bias Error on the Calibration Error of a High-Precision RLG INS System-Level Fitting Method. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 9511009. [CrossRef]
16. Lu, J.; Ye, L.; Niu, Z.; Dong, J.; Su, A. An all-parameter calibration for 6-axis skewed IMU. *IEEE Trans. Ind. Electron.* **2023**, *70*, 3126–3135. [CrossRef]
17. Pan, J.; Li, B.; Zhang, H.; Gao, F.; Li, X.; Zhou, G. Self-Calibration of INS/Odometer Integrated Navigation System With a Large Initial Position Error. *IEEE Sens. J.* **2024**, *24*, 11048–11056. [CrossRef]
18. Cheng, J.; Liu, J.; Cai, J.; Xu, Y. A Polar Integrated Alignment Assisted by DVL Under Large Azimuth Misalignment. *IEEE Sens. J.* **2023**, *23*, 5962–5973. [CrossRef]
19. Wang, Q.; Liu, K.; Cao, Z. System noise variance matrix adaptive Kalman filter method for AUV INS/DVL navigation system. *Ocean Eng.* **2023**, *267*, 113269. [CrossRef]
20. Liu, C.; Wang, J.; Cao, J. Multi-resolution Noise Reduction Models for Denoising INS Raw Data. *Energy Procedia* **2011**, *13*, 6677–6683.
21. Zhi, Z.; Liu, D.; Liu, L. A performance compensation method for GPS/INS integrated navigation system based on CNN-LSTM during GPS outages. *Measurement* **2022**, *188*, 110516. [CrossRef]
22. Abdolkarimi, E.; Mosavi, M. A modified neuro-fuzzy system for accuracy improvement of low-cost MEMS-Based INS/GPS navigation system. *Wirel. Pers. Commun.* **2023**, *192*, 1369–1392. [CrossRef]
23. Wang, D.; Xu, X.; Yang, Y.; Zhang, T. A quasi-Newton quaternions calibration method for DVL error aided GNSS. *IEEE Trans. Veh. Technol.* **2021**, *70*, 2465–2477. [CrossRef]
24. Lv, P.; He, B.; Guo, J.; Shen, Y.; Yan, T.; Sha, Q. Underwater navigation methodology based on intelligent velocity model for standard AUV. *Ocean Eng.* **2020**, *202*, 107073. [CrossRef]
25. Yao, Y.; Xu, X.; Hou, L.; Deng, K.; Xu, X. A simple and precise correction method for DVL measurements under the dynamic environment. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10750–10758. [CrossRef]
26. Xu, B.; Guo, Y. A novel DVL calibration method based on robust invariant extended Kalman filter. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9422–9434. [CrossRef]
27. Zhao, L.; Liu, X.; Wang, L.; Zhu, Y.; Liu, X. A pretreatment method for the velocity of DVL based on the motion constraint for the integrated SINS/DVL. *Appl. Sci.* **2016**, *6*, 79. [CrossRef]
28. Ansari-Rad, S.; Hashemi, M.; Salarieh, H. Pseudo DVL reconstruction by an evolutionary TS-fuzzy algorithm for ocean vehicles. *Measurement* **2019**, *147*, 106831. [CrossRef]
29. Yona, M.; Klein, I. MissBeamNet: Learning missing Doppler velocity log beam measurements. *Neural Comput. Appl.* **2024**, *36*, 4947–4958. [CrossRef]
30. Klein, I.; Lipman, Y. Continuous INS/DVL fusion in situations of DVL outages. In Proceedings of the 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), Newfoundland and Labrador, St. John's, NL, Canada, 30 September–2 October 2020.
31. Kinsey, J.; Whitcomb, L. In situ alignment calibration of attitude and Doppler sensors for precision underwater vehicle navigation: Theory and experiment. *IEEE J. Ocean. Eng.* **2007**, *32*, 286–299. [CrossRef]

32. Wang, G.; Xu, X.; Yao, Y. An iterative Doppler velocity log error calibration algorithm based on newton optimization. *Math. Probl. Eng.* **2020**, *2020*, 3194034.
33. Li, D.; Xu, J.; Zhu, B.; He, H. A calibration method of DVL in integrated navigation system based on particle swarm optimizatio. *Measurement* **2022**, *187*, 110325. [CrossRef]
34. Cohen, N.; Klein, I. BeamsNet: A data-driven approach enhancing Doppler velocity log measurements for autonomous underwater vehicle navigation. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105216. [CrossRef]
35. Braginsky, B.; Baruch, A.; Guterman, H. Correction of DVL error caused by seafloor gradient. *IEEE Sens. J.* **2022**, *20*, 11652–11659. [CrossRef]
36. Tal, A.; Klein, I.; Katz, R. Inertial navigation system/doppler velocity log (INS/DVL) fusion with partial DVL measurements. *Sensors* **2017**, *17*, 415. [CrossRef] [PubMed]
37. Liu, Y.; Luo, Q.; Zhou, Y. Deep Learning-Enabled Fusion to Bridge GPS Outages for INS/GPS Integrated Navigation. *IEEE Sens. J.* **2022**, *22*, 8974–8985. [CrossRef]
38. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning: ICML 2013, Atlanta, GA, USA, 16–21 June 2013.
39. Armaghani, D.; Asteris, P. A comparative study of ANN and ANFIS models for the prediction of cement-based mortar materials compressive strength. *Neural Comput. Appl.* **2021**, *33*, 4501–4532. [CrossRef]

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.