

Article

Data-Driven Prediction of Experimental Hydrodynamic Data of the Manta Ray Robot Using Deep Learning Method

Jingyi Bai ^{1,2}, Qiaogao Huang ^{1,2,*}, Guang Pan ^{1,2} and Junjie He ^{1,2} 

¹ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China

² Key Laboratory for Unmanned Underwater Vehicle, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: huangqiaogao@nwpu.edu.cn; Tel.: +86-13991288604

Abstract: To precisely control the manta ray robot and improve its swimming and turning speed, the hydrodynamic parameters corresponding to different motion control variables must be tested experimentally. In practice, too many input control parameters will bring thousands of groups of underwater experiments, posing challenges to the duration and operability of the engineering project. This study proposes a generative adversarial network model to reduce the experimental period by predicting the hydrodynamic experimental time-series data of forces and torques in the three-coordinate directions in a Cartesian coordinate system through different combinations of motion control parameters. The motion control parameters include the rotation amplitude, frequency, and phase difference of the four steering gears which drive the pectoral fins. We designed the prototype and experimental platform and obtained 150 sets of experimental data. To prevent overfitting, the size of the dataset was expanded to 2250 groups by slicing time series, and the subsequences of varying lengths were extended to the same length by LSTM. Finally, the GAN model is used to predict the hydrodynamic time series corresponding to the different motion parameters. The results show that the GAN model can accurately predict the input both from the validation set and the unlearned interpolated motion parameters. This study will save experimental time and cost and provide detailed hydrodynamic experimental data for the precise control of manta rays.



Citation: Bai, J.; Huang, Q.; Pan, G.; He, J. Data-Driven Prediction of Experimental Hydrodynamic Data of the Manta Ray Robot Using Deep Learning Method. *J. Mar. Sci. Eng.* **2022**, *10*, 1285. <https://doi.org/10.3390/jmse10091285>

Academic Editor: Rafael Morales

Received: 19 July 2022

Accepted: 9 September 2022

Published: 12 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As a kind of small intelligent equipment, autonomous underwater vehicles (AUV), which can move underwater for a long time, are widely utilized in environment monitoring, biological research, shipwreck salvage, and so on. However, the traditional propeller-type underwater vehicle has low maneuvering efficiency, poor maneuverability, and high noise, which may reduce the endurance time and damage the ecological environment. In the meantime, long-term natural selection and evolution have endowed fish with extraordinary athletic abilities: low-energy long-distance cruising, powerful athletic explosiveness, maneuverability, etc. Therefore, more and more researchers seek inspiration in bionics to improve the performance of underwater vehicles [1–3]. The movement patterns of fish are divided into two modes: median and/or paired fin (MPF) locomotion and body and caudal fin (BCF) locomotion [4]. The manta rays in the MPF mode flap the pectoral fins to generate thrust, which can complete high-mobility actions such as zero-radius turning and instantaneous acceleration [5–7]. The bow-gliding motion mode of the manta ray also reduces energy consumption. Cai et al. designed a manta ray robot using pneumatic artificial muscles [8]. Zhou et al. designed a manta ray-like vehicle driven by six flexible fins, with a speed of up to 1 BL/S and a continuous operation of 90 hours [9].

In 2020, a bionic manta ray underwater robot was designed by researchers of our team, as shown in Figure 1. The vehicle has a wingspan of 2 m and a total mass of 100 kg. It

is driven by four steering gears to flap its wings, and its motion attitude is controlled by adjusting motion parameters such as the rotation amplitude, frequency, and phase difference between the steering gears. To optimize the motion control parameters of the vehicle, we need to obtain the hydrodynamic parameter values of the vehicle corresponding to different motion attitudes. There are usually two ways to do this, and the first is an accurate numerical simulation by the CFD method. Fish et al. analyzed manta rays' powerful lift generation mechanism through the immersion boundary method and estimated the maximum operating efficiency to be 89% [10]. Huang et al. established the motion equation of the manta ray vehicle with six degrees of freedom and calculated the influence of different pectoral fin flapping motion parameters on the propulsion force through the computational fluid dynamics method [11]. The second is to test the hydrodynamic parameters corresponding to different motion modes through experiments. Through experiments, Yang et al. tested the effects of different control parameters of the manta ray robot on propulsion speed and steering maneuverability [12]. However, both methods have shortcomings. The model of CFD simulation has errors compared to the actual vehicle because the simulation model cannot imitate the motion deformation, the internal flexibility distribution, and the surface friction of the real one entirely. On the other hand, the experiment cannot exhaustively test all combinations of input control variables due to the cost of time and labor.



Figure 1. The bionic manta ray robot driven by four steering gears.

Traditional fluid mechanics calculation methods are difficult to calculate for complex characteristic problems such as three-dimensional unsteady flow fields and multi-scale nonlinearity. There may be deviations between theoretical calculations and experimental results for complex engineering problems with high Reynolds numbers. At the same time, deep learning is an algorithm for feature identification, rule exploration, and data regression based on big data [13–16]. It performs excellently in high-dimensional nonlinear space and is currently widely used in health monitoring systems [17,18] and cyber-physical systems [19]. Since CFD research will generate a large amount of flow field data, which can effectively support neural network training, researchers began to use neural network methods to solve high-dimensional nonlinear problems in flow fields [20–22]. At present, deep learning is widely used in the prediction and optimization of hydrodynamic parameters [23–25] and physical fields [26–28], and is widely used in both theoretical analyses [29,30] and flow field experiments [31,32]. In past research, J P Panda et al. predicted the complex flow field over an axisymmetric rotation body [33] and the wind loads and damage detection of ships [34]. Yang et al. constructed the resistance value prediction model corresponding to the different draft depths of the ship model through the BFRNN algorithm [35]. Using deep learning for building mechanics research has two significant advantages. First, the parameter terms in the partial differential equation of the flow field can be optimized through neural networks to improve the accuracy of CFD [36,37]. Secondly, it provides an instant and reliable solution to some complex engineering flow problems with high Reynolds that are difficult to solve by CFD methods [38]. Therefore, for the problem of obtaining the hydrodynamic parameters of a flapping manta ray robot,

we propose a deep learning framework to predict the force and torque corresponding to different motion parameter combinations.

The purpose of this research is to establish a generative adversarial neural network deep learning framework which can predict the hydrodynamic experiment time series data of force and torque in the three-coordinate direction in the Cartesian coordinate system through the combination of motion parameters such as different flapping amplitudes and frequencies of the bionic manta ray robot. The key contributions of this paper can be summarized as follows:

- NuWe made a scaled-down experimental model of the bionic vehicle and built a platform for collecting experimental data.
- We effectively expanded the data sample size by slicing the time series and supplementing the subsequences to the same sequence length.
- We utilized the GAN model to predict the hydrodynamic parameters corresponding to the input values of the validation set and interpolation variables, respectively, and compared the predicted results with the experimental results to verify the accuracy of the results.

The work of this paper will be introduced from the following aspects: the second part introduces the hydrodynamic experimental test results, data set augmentation, and data normalization methods. The third part lists the different generator and discriminator models for predicting hydrodynamic parameters. The fourth part provides the prediction results of different data augmentation methods, model combinations, and input variables. The last part provides conclusions of the research.

2. Datasets

2.1. Experimental Prototype Model

As shown in Figure 1, the original manta ray robot prototype has a wingspan of 2 m and a weight of 100 kg. It is not easy to directly use the original prototype for experiments because multiple people are required to cooperate in the work, which involves lifting, assembling, and filling the buoyancy block. Due to the size of the original prototype, the manta ray robot needs to flap at a depth of 2–3 m underwater. In addition, the results corresponding to hundreds of different input motion combinations should be measured to obtain the whole hydrodynamic parameter time series datasets. Using the original prototype will consume a lot of workforce and time. Therefore, in this paper, a scaled-down model of a bionic vehicle driven by four steering gears is fabricated, and a six-dimensional force sensor is equipped at the center of the model's mass. The size comparison between the model prototype and the original prototype is shown in Table 1. An experimental platform was constructed to conduct scale model experiments, as shown in Figure 2. The main body of the platform is a water tank with the size of 1000 × 600 × 600 (mm), reinforced with aluminum profiles.

Table 1. Size comparison between the experimental prototype and the actual prototype.

	Experimental Prototype	Actual Prototype
Chord (mm)	200	1000
Span (mm)	400	2000
Thickness (mm)	60	300

The model prototype is fastened to the profile frame through the threaded connection parts on the force sensor at its center of mass. Some buoyancy blocks are placed in the prototype to make the buoyancy in the water equal to gravity. The control module of the steering gears and the signal receiver of the six-dimensional force sensor are placed outside the water tank and connected to the computer. Furthermore, we use the STM32 microcontroller (STMicroelectronics Inc., Geneva, CH) as the control module of four steering gears and the Nano17 six-dimensional force sensor (ATI Inc., North Carolina, NC, USA) as the force measurement and data acquisition module. During the experiment, the upper

computer software system instructed the drive board to control each steering gear to swing with different amplitudes, frequencies, and phase differences. During the flapping process, the six-dimensional force sensor will acquire the force and moment information in three directions in the Cartesian coordinate system in real-time. The information is transmitted to the computer through the force and torque sensor system of Nano17 (ATI Inc, North Carolina, NC, USA), and we can obtain the hydrodynamic data corresponding to a set of motion parameters. We need to measure the hydrodynamic parameter curve for each set of motion control parameters with a time step of 3000. The detailed data acquisition process of the experiment is shown in Figure 3.

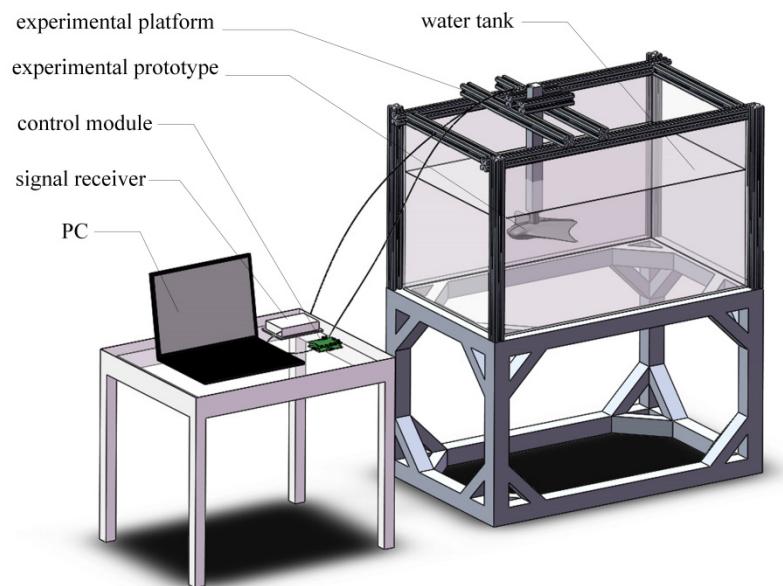


Figure 2. Schematic diagram of the experimental platform: The experimental device includes a water tank, an aluminum profile frame, a scaled model prototype, a steering gear control module, a six-dimensional force signal receiver, and a computer. The scaled model prototype is fixed in the water tank through the sensor screw connection at the centroid position.

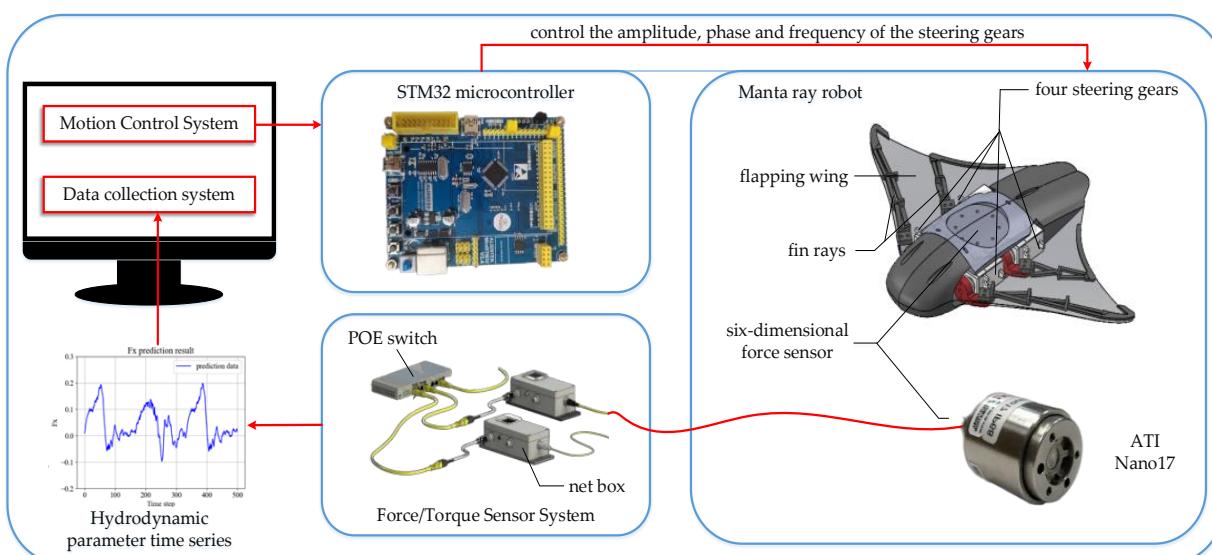


Figure 3. The detailed data collection process of the experiment. (The upper computer software system sent instructions to the STM32 microcontroller to drive each steering gear. During the wing flapping, the ATI Nano17 sensor acquires the force information and transmits it to the computer by the force sensor system.).

2.2. Data Obtained from Hydrodynamic Experiments

In the experiment, we tested the six-dimensional force parameters, namely lift, drag, lateral force, and their respective moments of the experimental prototype under different flapping amplitudes, frequencies, and phase differences. Among them, single-wing and double-wing flapping amplitudes are measured in a set of data every 5° , and the test amplitude ranges from 20° to 40° . The flapping frequency is measured in a set of data every 0.2 Hz, and the test frequency ranges from 0.3 Hz to 0.7 Hz. The phase difference between the front and rear fin rays is measured every 10° , and the test phase difference ranges from 0° to 40° . Based on the above input motion parameters, a total of 150 sets of experimental conditions were tested as the subsequent training set and validation set data. The nomenclature of the motion parameters is defined in Table 2, and some of the tested motion parameters are shown in Table 3. Part of the dataset was uploaded to GitHub: <https://github.com/lingke12138/prediction-of-experimental-data.git> (accessed on 11 May 2022).

Table 2. The symbols of motion parameters.

Motion Parameters	Amplitude Left	Amplitude Right	Frequency	Phase Difference
symbols	A_L	A_R	f	Φ

Table 3. Part of the test motion parameters.

Parameters	A_L ($^\circ$)	A_R ($^\circ$)	f (Hz)	Φ ($^\circ$)
Input Data	20	20	0.3	0
	20	25	0.3	0
	20	20	0.5	0
	20	20	0.5	10
	20	25	0.5	10

In addition, we conducted additional experiments on the interpolation of variable input combinations (A_L, A_R, f, Φ) and used the results as a test set for our deep learning method. We set up four sets of input data: (33, 33, 0.7, 40), (20, 33, 0.7, 40), (40, 40, 0.6, 40) (40, 40, 0.7, 25), which represent four experimental conditions of symmetric and asymmetric flapping with variable amplitude, symmetric flapping with variable frequency, and flapping with variable phase difference.

In the experimental hydrodynamic test, the six-dimensional force sensor is set to sample 100 times per second, and the sampling time of each set of data is 30 s. The total time step of experimental sampling is 3000 steps, and the format of data obtained from the experiment is shown in Table 4. The variation law of the hydrodynamic parameters corresponding to the motion parameters is shown in Figure 4. Part of the data set has been uploaded to GitHub, including four kinds of motion conditions: symmetric flutter, asymmetric flutter, variable amplitude, and variable phase difference.

Table 4. Experimental data of $A_L = A_R = 40, f = 0.5, \Phi = 30$.

Step	F_x	F_y	F_z	T_x	T_y	T_z
1	0.087	0.007	-0.795	-5.341	-15.710	-2.114
2	0.084	0.012	-0.836	-6.324	-16.279	-2.802
3	0.092	0.005	-0.829	-6.898	-16.952	-2.673
4	0.091	-0.002	-0.850	-7.858	-17.377	-2.471
...
2998	0.081	-0.003	-0.729	-4.959	-13.400	-0.920
2999	0.080	0.001	-0.770	-5.029	-15.104	-1.541
3000	0.085	-0.003	-0.803	-5.881	-16.196	-1.664

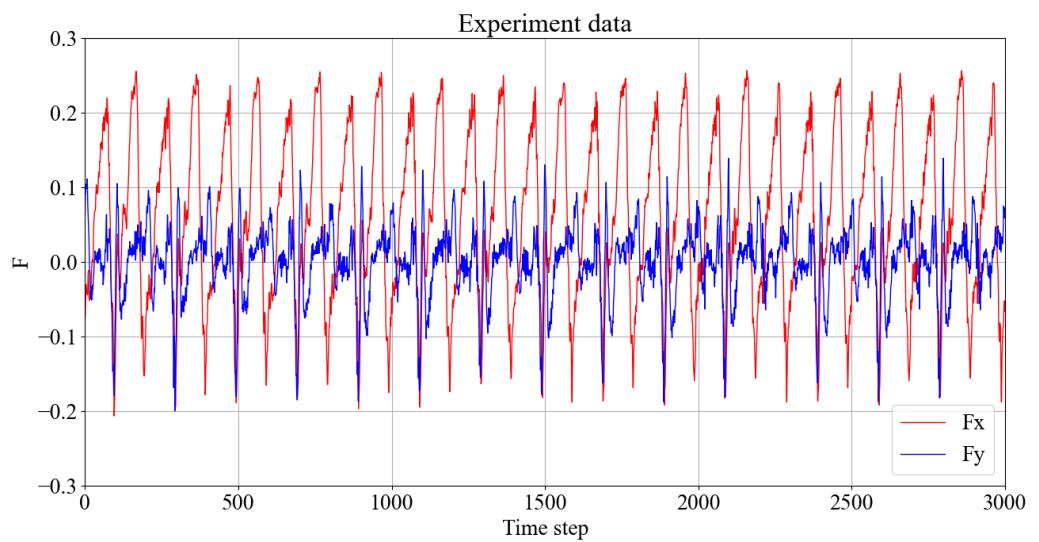


Figure 4. Time-series of hydrodynamic parameters obtained by experiments directly with the input of $A_L = A_R = 40, f = 0.5, \Phi = 30$.

To prove that the data obtained by the scaled-down model test is also valid for the original prototype, we conducted a flapping experiment of the original prototype with input parameters of $A_L = A_R = 40, f = 0.5, \Phi = 30$ and compared the experimental results. The experimental test of the original and scaled-down manta ray robot is shown in Figure 5. The comparison of the forces in the x and z directions of the two robots is shown in Figure 6. The forces in all directions have the same change trend, and the force is proportional to the prototype's size. Therefore, the results obtained from the scaled model experiment can be used as a reference for the original prototype.



Figure 5. The experimental test photos of the original and scaled-down manta ray robot: (a) original manta ray robot with a span of 2000 mm, (b) scaled-down manta ray robot with a span of 400 mm.

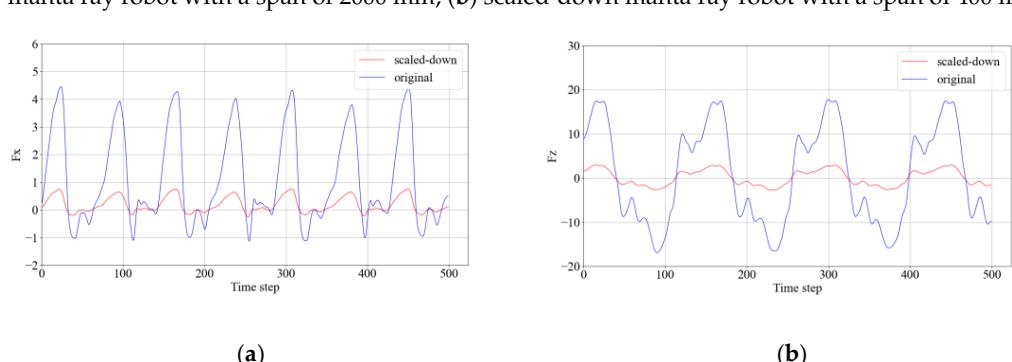


Figure 6. F_x and F_z of scaled-down and original manta ray robot measured with input values of $A_L = A_R = 40, f = 0.5, \Phi = 30$. (a) the comparison of F_x , (b) the comparison of F_z .

2.3. Datasets of Deep Learning

Through the hydrodynamic experiments, we obtained 150 sets of data. However, for neural network training, the amount of data is insufficient and will cause an overfitting phenomenon, which means the neural network fits the training set data too much and has poor generalization ability to other input data. Data augmentation is a technique of generating grammatical data to solve insufficient data samples for training and to obtain better prediction accuracy without collecting data. This can effectively solve the problem of overfitting. There are three traditional data enhancement methods for solving sequence data: time offset, amplitude-frequency conversion, and noise injection. It aims to close the gap between training samples and test data by expanding the sample size. At the same time, by the hydrodynamic experiments, we obtained multi-period force and moment data with the time step of 3000. Therefore, this study intends to expand the data set by slicing the time series obtained by the force sensor and extracting complete flapping period data each time.

Since there are slight numerical fluctuations during each flutter cycle of the measured data, the method of slicing the data according to the cycle is equivalent to adding noise to each group of data series, and the data can be enhanced by this method. Due to the experimental conditions of flapping by variable frequency, there are differences in the data lengths sliced by each period. For example, when flapping at a frequency of 0.7 Hz, there are 143 sampling time steps per cycle, but when flapping at 0.3 Hz, there are 333 sample time steps. To facilitate the operation of the neural network generator, such as the subsequent convolution, each experimental data sequence is extended to 500-time steps in this study to ensure the consistency of sequence length. The data of 500-time steps are not sliced from the original data directly because this method will generate a large amount of repeated data. Especially when beating at 0.7 Hz, the second cycle in the sequence data is wholly duplicated. Since too many mutations in the data will affect the accuracy of the prediction, the original experimental data of this paper is smoothed by the method of Savitzky-Golay to reduce noise. The original and smoothed data are shown in Figure 7. After that, two methods are verified to expand the sequence after the slice: one is to add 0 to the data directly, and the other is to predict the time-series changes of sliced data utilizing the long short-term memory neural network. The latter method takes the front 30 data points of the sequence as input to predict the following 30 data points and predicts the subsequent data changes of each slice using a loop program. The augmented dataset sequence of the two methods is shown in Figure 8. This method expands the 150 sets of working conditions in this experiment to 2250 datasets.

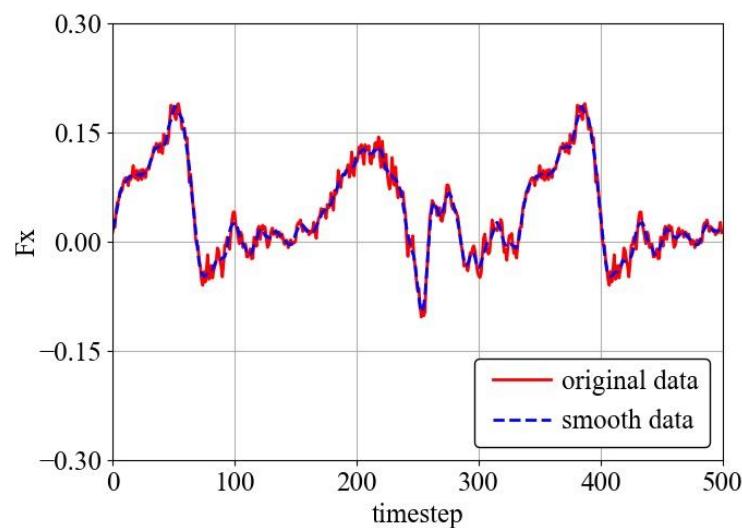


Figure 7. The original and smoothed data.

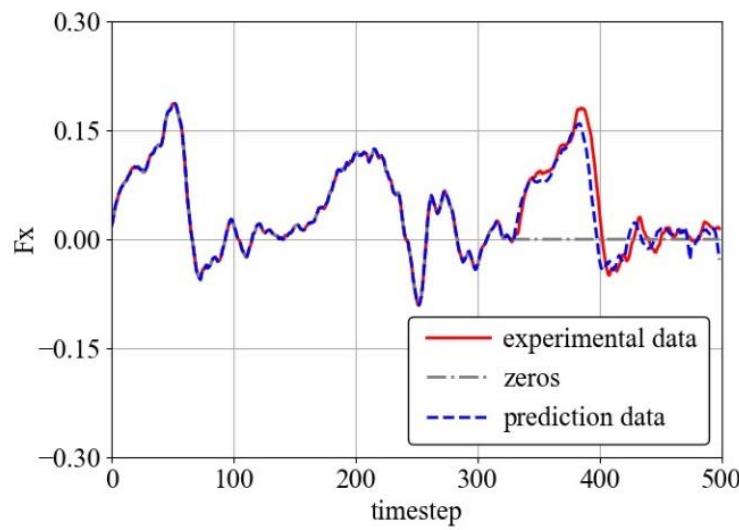


Figure 8. Data of expand to length 500 by padding with zeros and prediction data.

As shown in Table 4, the data range of F and T is quite different, and the values of T_y are several times those of F_x . Feeding the original data directly into the neural network for training will seriously affect the prediction accuracy of F_x . Therefore, the Min-Max scaler normalization method, shown below, is utilized to normalize the data set so that the data is linearly scaled to a range of $(0, 1)$.

$$\begin{aligned} x_{std} &= \frac{x - x_{min}}{x_{max} - x_{min}} \\ x_{scaled} &= x_{std} * (max - min) + min \end{aligned} \quad (1)$$

where, x_{std} is scaling ratio, (min, max) is scaled feature range, x_{scale} is data after normalization. The original time series of F_x and T_y of the same input is shown in Figure 9a, and the normalized data is shown in Figure 9b.

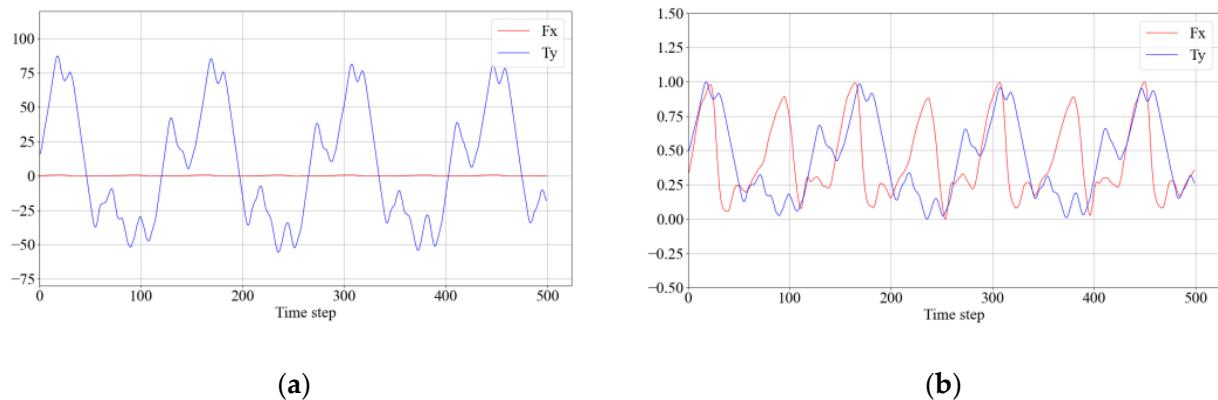


Figure 9. (a) The original time series data of F_x and T_y , (b) The normalized time series data of F_x and T_y .

Finally, the 2250 sets of data are randomly scrambled and divided into training and validation sets at the ratio of 8:1.

3. Methods

3.1. Conceptual Figure and Algorithm Selection

Prediction of the experimental hydrodynamic data of the manta ray robot mainly includes three steps: the acquisition of experimental data, training, and prediction of the GAN model, and error analysis by comparing the experimental data. The detailed conceptual flow chart is shown in Figure 10.

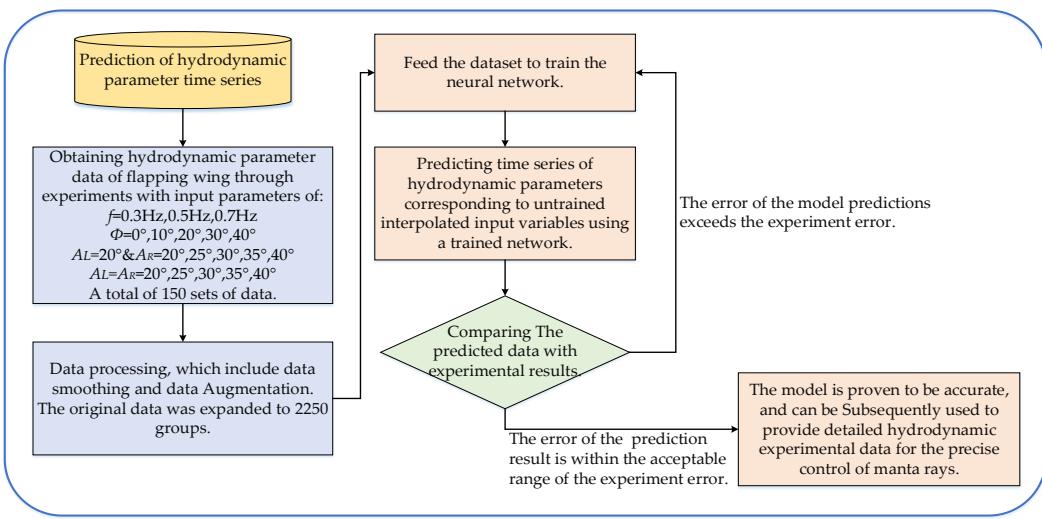


Figure 10. The detailed conceptual figures of hydrodynamic parameter time series prediction include experimental data acquisition (hydrodynamic experiments and data augmentation), deep learning (training the GAN model and predicting the time series corresponding to different input parameters), and error analysis (determining whether the prediction error is less than the experimental error range).

The blue boxes represent data acquisition steps, the orange boxes represent training models and predictions, and the green boxes represent error analysis. We have discussed the data acquisition in detail in Chapter 2 and can find from it that the input variables which will feed to the neural network are $\{A_L, A_R, f, \Phi\}$, while the output variables are $\{F_x, F_y, F_z, T_x, T_y, T_z\}$. Therefore, the neural network is trained to obtain a nonlinear mapping of motion control variables to hydrodynamic parameter time series, which can be summarized in Equation (2):

$$y = g(F_x, F_y, F_z, T_x, T_y, T_z) = f(\lambda_m^k, A_L, A_R, f, \phi) \rightarrow \hat{y} \quad (2)$$

among them, y is the prediction output sequences, \hat{y} is the data obtained from the experiments, and λ_m^k is trainable parameters of each hidden layer nodes. Deep learning is continuously learning to optimize the trainable parameters in the neural network so that the prediction results y are constantly close to the actual value \hat{y} .

In this study, the neural network model we built uses three basic neural network frameworks, namely the full connection neural network, the convolutional neural network, and the transposed convolutional neural network.

The algorithm of the full connection neural network is to solve the logistic regression of the mapping between each neuron in each layer, as shown in the Formula (3):

$$y^i = \sum_j w^{j,i} x^j + b \quad (3)$$

where, $w^{j,i}$ is a weight parameters tensor between input(x) and output(y) neurons, and b is a constant of bias. The number of neurons of input and output layers is defined as j and i .

The convolution calculation generally uses an n-dimensional convolution kernel which slides on the input tensor according to the specified step size and traverses each feature point on the tensor. Since the input and output data in this paper are all digital sequences, 1d convolution is chosen. The mapping between the input and output of the 1d convolutional layer can be expressed by the Formula (4):

$$y_j = \left[\sum_i \sum_{n=0}^{N-1} w_n^i x_{j+n}^i \right] + b \quad (4)$$

where, w is the weight parameters tensor of a convolution layer in kernels, and b is a constant, which is also a parameter to be learned. Kernels are one-dimensional tensors, and their size is defined as N .

The transposed convolutional neural network, also known as deconvolution, can restore the sequence or image dimensions before the convolution operation according to the size of the convolution kernel and the size of the output. As the operation of standard convolution, transposed convolution optimizes the trainable variables through the neural network to obtain optimal output results.

It can be seen that the above neural networks are all linear calculations. To improve the neural network's ability to fit data, researchers generally apply an activation function to the convolutional or fully connected layer to provide a nonlinear factor for the neural network. In this paper, we use three types of activation functions $f(x)$: *ReLU*, *tanh*, and *sigmoid*. They are respectively defined as:

$$\begin{aligned} \text{ReLU} : f(x) &= \max(0, x) \\ \tanh : f(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \text{Sigmoid} : f(x) &= \frac{1}{1+e^{-x}} \end{aligned} \quad (5)$$

The deep learning model GAN is a promising unsupervised learning method in recent years and has been widely used in image generation and data enhancement. GAN consists of two networks: a generator model and a discriminator model. The generator model is used to generate a series of sequences, while the discriminator model is used to determine the category of the above-generated sequence or calculate the error between the generated and actual sequence. The structure of the deep learning networks in the present study is shown in Figure 11.

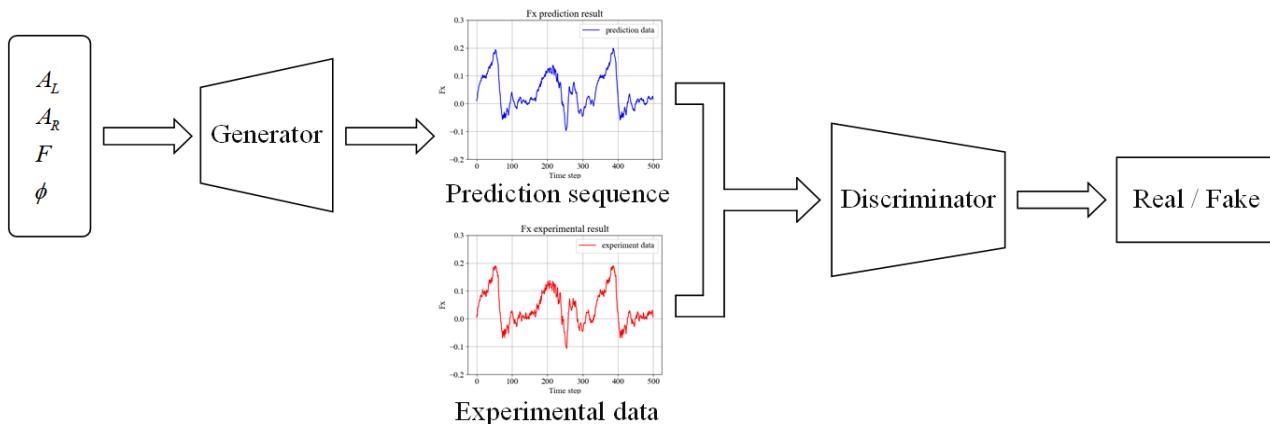


Figure 11. The structure of the deep learning networks in the present study.

3.2. Generative Adversarial Networks

In the generator model selection, this study verifies the prediction effect of the fully connected network and the transposed convolutional neural network and defines them as G1 and G2, respectively. The number of hidden layer nodes of the model G1 is (10, 20, 100, 300, 500). Except that the last layer is *tanh*, the activation function of each hidden layer selects *ReLU*. The model G2 consists of four Conv1d_transpose layers, a flatten layer, and a dense layer. The activation functions of the transposed convolutional layers are all *ReLU*, and the fully connected network is *tanh*. The generator models G1 and G2 are shown in Figure 12.

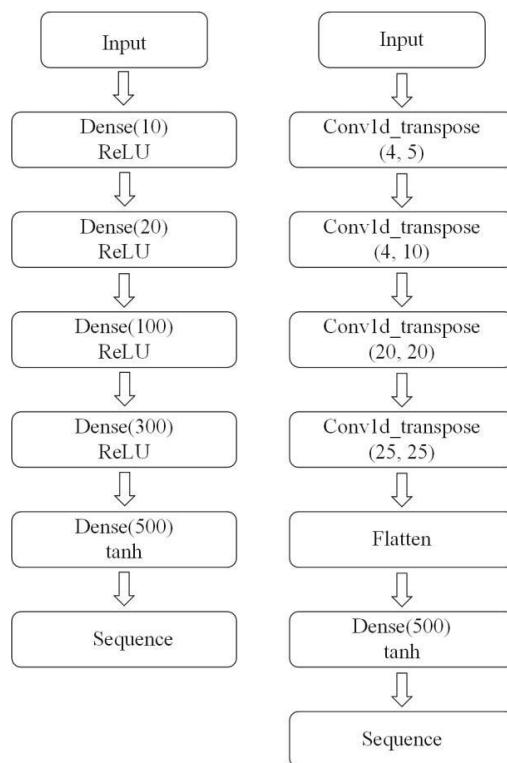


Figure 12. Generator network model of fully connection and transposed convolution methods.

In the discriminator model selection, a fully connected neural network and a convolutional neural network are verified in the present study and are defined as D1 and D2, respectively. The number of hidden layer nodes of model D1 is (100, 80, 50, 20, 1), and the activation function is Sigmoid in the last hidden layer, with the others being *ReLU*. The model D2 includes three Conv1d_transpose layers whose activation functions are *ReLU* and a Dense layer whose activation function is Sigmoid. The discriminator models D1 and D2 are shown in Figure 13.

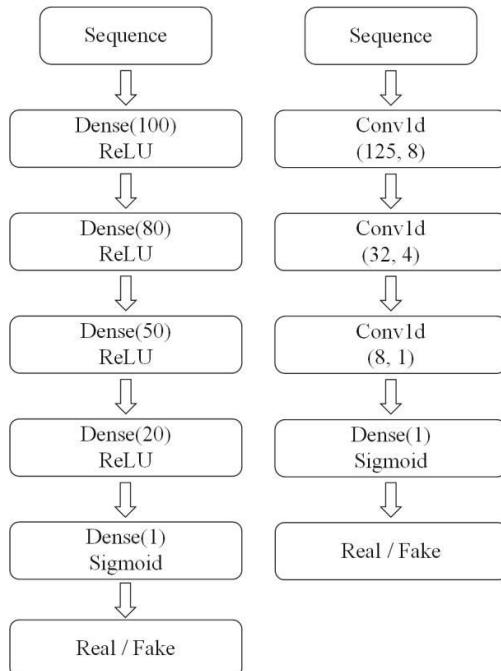


Figure 13. Discriminator network model of full connection and convolution methods.

During the training process, the weight parameters set of the generator and discriminator model are optimized by the Adam optimizer, and the initial learning rate is set to 0.001 with the gradient decay rate of 0.9. The selection of hyperparameters for each neural network is detailed in Appendix A, which includes the number of hidden layer nodes of G1 and D1, the kernel size of G2 and D2, and the learning rate of the optimizer. The loss function of the generator is mean squared error, and the loss function of the discriminator is binary cross entropy, defined as Formulas (6) and (7), respectively.

$$Loss_m = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - f(x^{(i)}))^2 \quad (6)$$

$$Loss_b = \frac{1}{n} \sum_{i=1}^n y^{(i)} \cdot \log(f(x^{(i)})) + (1 - y^{(i)}) \cdot \log(1 - f(x^{(i)})) \quad (7)$$

where n is the number of samples, $y^{(i)}$ is the actual value, and $f(x^{(i)})$ is the prediction value of the neural network.

In the model validation part, the evaluation methods such as mean squared error (MSE) and mean absolute error (MAE) are used to verify the performance of GAN models. The MAE evaluation method is defined in Formula (8):

$$MAE = \frac{1}{n} \sum_{i=1}^n |(y^{(i)} - f(x^{(i)}))| \quad (8)$$

The training datasets are divided into batches of 32, and the number of epochs required for the network training is discussed in the next chapter of the present study. GAN models are trained on the Tensorflow framework, and by an NVIDIA GeForce RTX2080 with a Max-Q design graphics processor.

4. Results

4.1. Results of Different Data Augmentation Methods:

In this section, the GAN model with the G1 + D1 network structure mentioned above is used to verify the prediction effect on the datasets with different extension methods mentioned in the second chapter. The different data sets are data filled with zero and data filled by the LSTM prediction method. The result of the input variable (A_L, A_R, f, Φ) equal to (40, 40, 0.3, 30) in the validation set is selected to show the prediction effect. The validation loss of two data extension methods is shown in Figure 14. The total epoch of this verification is 50,000, and the prediction result of F_x is autosaved every 100 epochs to check whether the neural network model has converged. The prediction results trained by 2000, 10,000, and 25,000 epochs are shown in Figure 15.

As shown in Figures 14 and 15, data with zero padding have a more minor mean square loss on the validation set. However, more training steps do not mean more accurate results on this dataset. It is obvious that when training for 2000 epochs, the zero-padding sequence can be fitted while the sequence padded with prediction data still has errors, which means the training speed with zeros-padding data is faster. However, as the number of training steps increases, the zero-padding sequence fitting results begin to deteriorate, and the loss on the validation set begins to increase. In addition, no matter how many epochs are trained, the zero-padding sequence always has some errors at the time steps of the initial position of zero-padding, which will affect the optimization of the loss function. Therefore, we select the sequence filled with prediction data for subsequent verification and prediction because of its good model stability and data fitting accuracy.

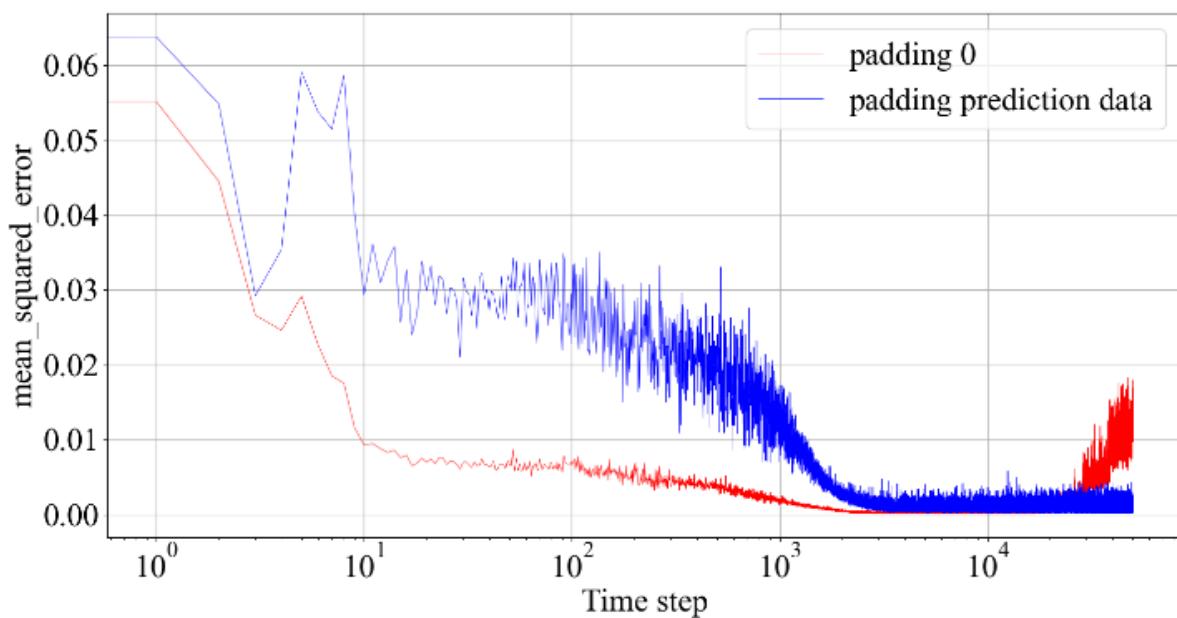


Figure 14. Validation loss of two data augmentation methods.

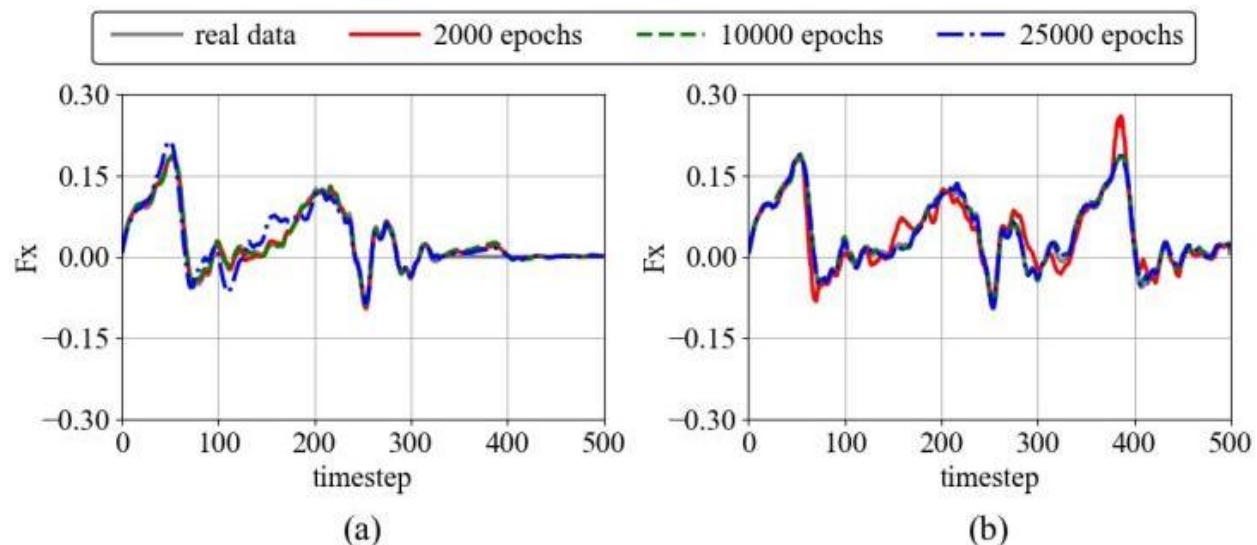


Figure 15. Prediction results of F_x with different epochs: (a) padding with zeros. (b) padding with prediction data.

4.2. Results of Different GAN Models

The prediction effect of different generators and discriminators mentioned in Chapter 3 are verified in this section. We use two evaluation methods of MSE and MAE to verify the excellence of the proposed model. The results are shown in Figure 16.

As Figure 16 above shows, the model tends to overfit when using the fully connected neural network G1 or the transposed convolutional neural network G2 alone for prediction. The final converged mean squared error and mean absolute error values corresponding to the four GAN models are close. However, the G2 + D2 model, which combines the transposed convolutional generator and convolutional discriminator model, has the best convergence. Therefore, the follow-up prediction research of this study will also adopt the G2 + D2 network model. For the possible overfitting problem, the early stopping method is used here, that is, the neural network training is terminated early at the end of 2000 epochs when training is stable.

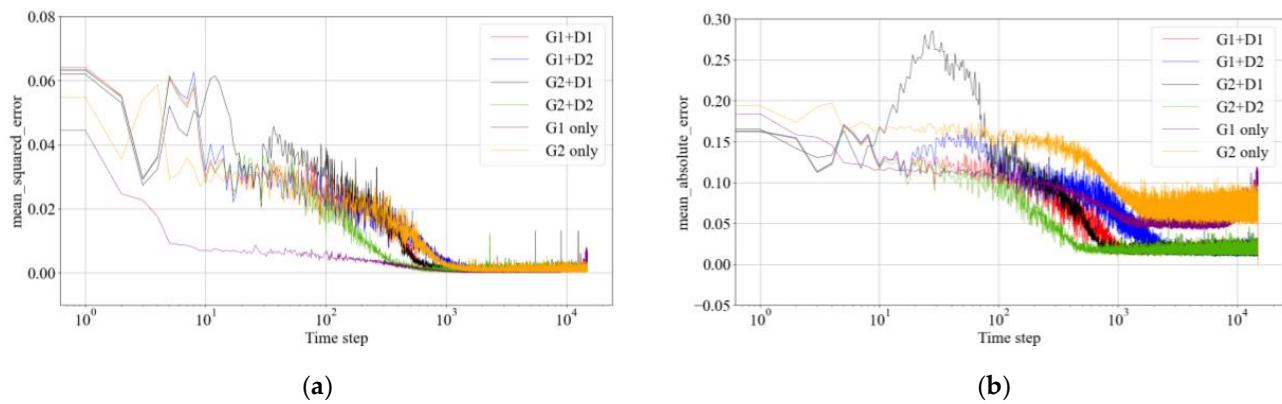


Figure 16. The performance of GAN models is verified by two evaluation methods. (a). MSE method. (b). MAE method. The models G1 and G2 represent generators of fully connective and transpose convolutional neural networks, while D1 and D2 represent discriminators of fully connective and convolutional neural networks.

4.3. Results of Different Input Motion Parameters

In this section, we predict the hydrodynamic timing sequence of the four types of motion modes of different input parameters of the manta-ray robot, namely the symmetrical and asymmetrical flapping of pectoral fins with variable amplitude, the symmetrical flapping of pectoral fins with variable frequency, and the symmetrical flapping of pectoral fins with a variable phase difference of front and rear steering gears. The prediction effects on the validation and test sets are shown, respectively. Among them, the displayed validation set data is obtained from the validation set of the neural network. Each of the four motion modes extracts one set of input parameters and visualizes the prediction results to verify the training accuracy of the model. The input of the test set is a combination of motion control variables (AL, AR, f, Φ). As mentioned in Section 2.1, the values of the input are $(33, 33, 0.7, 40)$, $(20, 33, 0.7, 40)$, $(40, 40, 0.6, 40)$, $(40, 40, 0.7, 25)$ representing the four motion modes respectively. The output of the test set is hydrodynamic time series obtained through experiments. In addition, the test data set has never been trained by a neural network.

For the experimental condition of pectoral fins flapping with variable amplitude, we fix (f, Φ) as $(0.7, 40)$, and in the case of asymmetrical flapping, the left amplitude A_L is fixed as 20. The input amplitude of the validation data is set to 25, and the test data is set to 33, respectively. The six-dimensional hydrodynamic time series is predicted through the neural network of G2 + D2 mentioned in Section 4.2. For the variable frequency case, the input variables (A_L, A_R, Φ) are fixed as $(40, 40, 40)$, and the variable f is set to 0.5 and 0.6, representing the validation and test data, respectively. For the case of variable phase difference, we fix (A_L, A_R, f) as $(40, 40, 0.7)$ and set the input parameter of the variable Φ to 10 for the validation set and 25 for the test set. The comparison between the prediction results on the validation set and test set data of the four experimental conditions and the experimental results are visualized to show the error of the prediction results. For the sake of clarity, the first 250 time steps of data are cut from the original data sequences, and the new sequence length of 250 can include the entire period of data for all flapping frequencies. The prediction results of the validation set are shown in Figure 17, and the prediction results of the test set are shown in Figure 18.

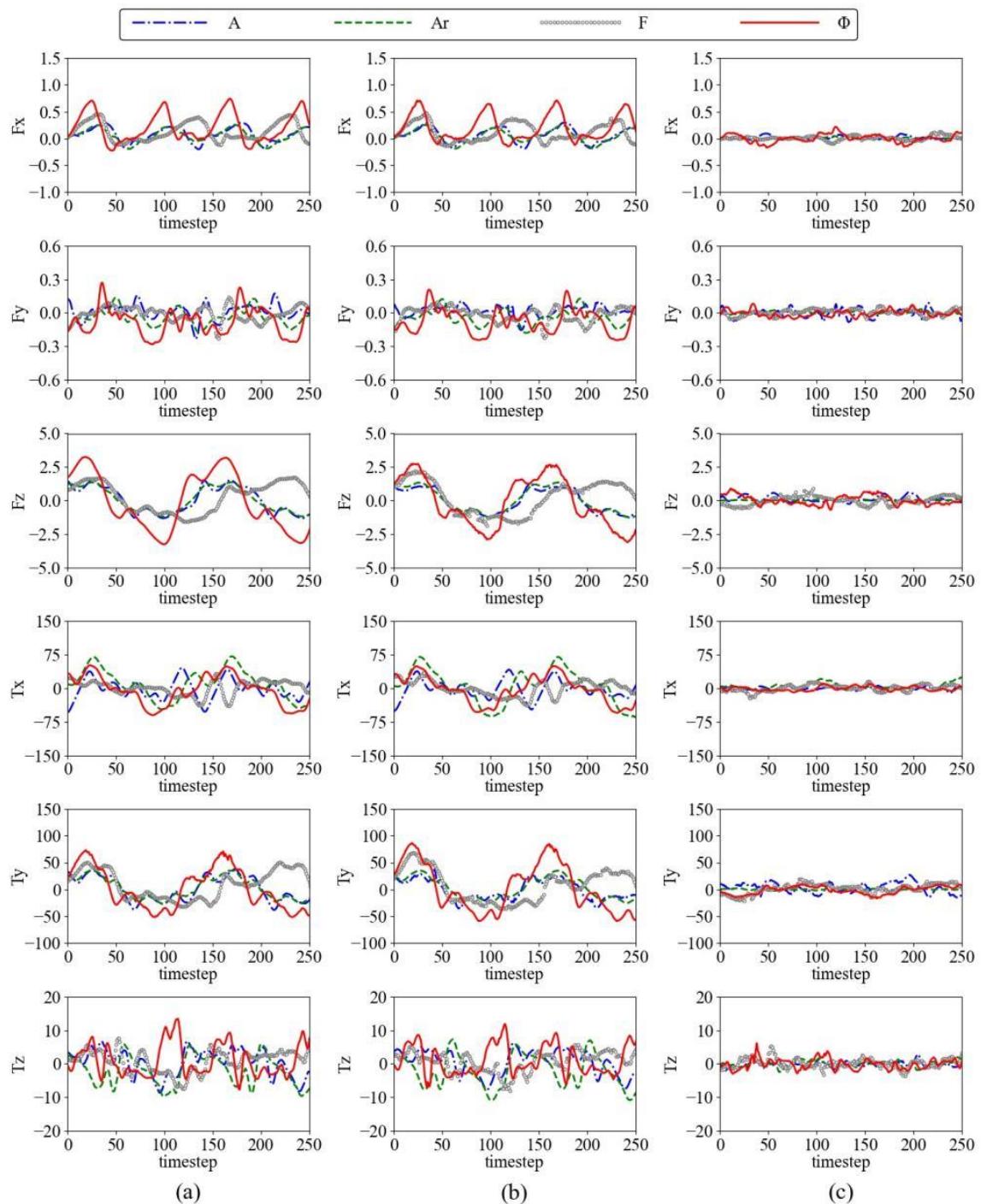


Figure 17. Comparison of the validation set prediction results with the experimental hydrodynamic data: (a) experimental hydrodynamic time series. (b) prediction data of validation set. (c) The error of the data at each time step between the experimental and predicted sequences.

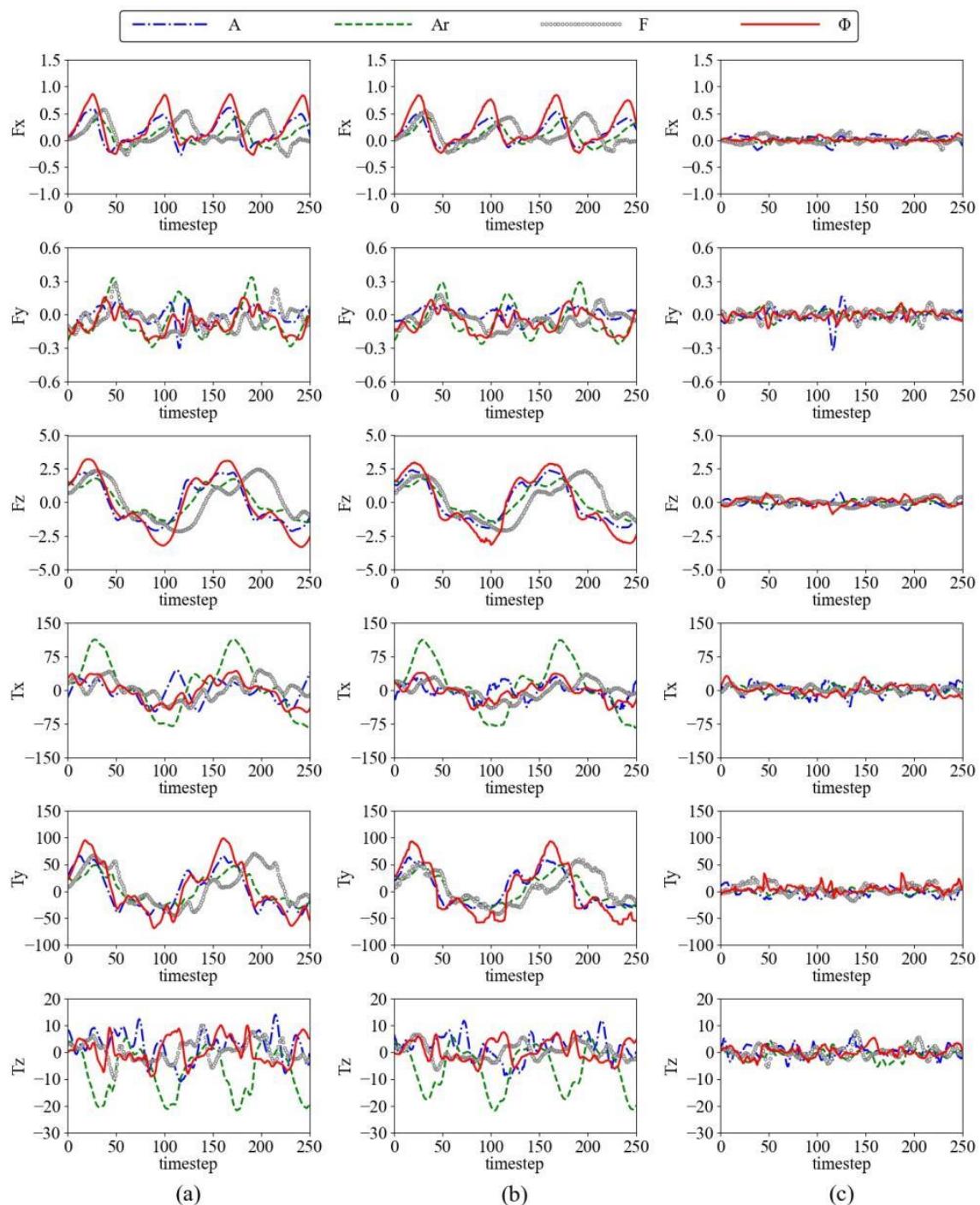


Figure 18. Comparison of the test set prediction results with the experimental hydrodynamic data: (a) experimental hydrodynamic time series. (b) prediction data of test set. (c) The error of the data at each time step between the experimental and predicted sequences.

The prediction results in Figure 16 show that the model performs well on the validation set, and the sequence of prediction results can fit the changing trend of the original experimental data set, indicating that the neural network model has certain prediction accuracy and model generalization. It can be seen from Figure 17 that the model also has a relatively accurate performance when migrating the interpolation input data that the neural network has not trained. Particularly for the smooth six-dimensional hydrodynamic data with no sudden change and evident periodicities, such as F_x and F_z , the neural network performs well and can accurately predict both the trend and extreme value of the time-series

data at the same time. For data with sharp gradients, such as F_y of the variable amplitude with the symmetric flapping condition and T_z of the phase difference condition, the model can predict the approximate change trend of the data, but has errors at the position of the extreme value in gradient mutation sequences. The F_x and T_z prediction results of variable phase difference conditions are utilized to represent the time series data with smooth and evident periodicity as well as the data with gradient mutation and insignificant periodicity. The error between the GAN model prediction results in the validation and the test set for these two types of data and the experimental data are calculated. Because some data are infinitely close to 0 in the sequence, which will cause excessive and abnormal error values when calculating the relative percentage error and affect the error evaluation, the absolute error is used here to judge the accuracy of the GAN model. The calculation formula is shown as Formula (9). In addition, one single cycle of data is extracted for each sequence to calculate the error. At the same time, we analyzed the data acquisition error during every single cycle of pectoral fin flapping in the experiment, which was calculated by Formula (10). The formulas are shown as follows:

$$E_{pred} = \frac{1}{m} \frac{1}{n} \sum_{j=1}^m \sum_{i=1}^n |x_{pred}^{j,i} - x_{test}^{j,i}| \quad (9)$$

$$E_{test} = \frac{1}{m} \frac{1}{n} \sum_{j=1}^m \sum_{i=1}^n |x_{test}^{j,i} - x_{ref,test}^{j,i}| \quad (10)$$

where, m and n represent the number of flapping periods in one original experimental sequence and the time steps number of each flapping period. $x_{pred}^{j,i}$, $x_{test}^{j,i}$, $x_{ref,test}^{j,i}$ represent the hydrodynamic variables of the prediction sequence, the experimental sequence of the j th period, and the reference experimental sequence, respectively.

The error of experiments and deep learning prediction are shown in Table 5. It can be seen that the error of the model prediction is within the acceptable range of the experiment errors. Therefore, the GAN model has confidence in predicting both smoothed and mutated hydrodynamic data.

Table 5. The absolute error of prediction and experiment.

	Validation Set	Test Set	Experiment
data smooth	0.0481	0.0434	0.0465
data with Gradient mutation	1.3264	1.4769	1.7249

5. Conclusions

In the present study, we first sliced and extended the hydrodynamic experimental data corresponding to the bionic manta ray robot flapping by different amplitudes, frequencies, and phase differences. In addition, the prediction accuracy and model convergence speed of various GAN structures were verified. Finally, we predicted the experimental hydrodynamic data corresponding to different interpolated motion parameter values and verified the prediction accuracy. The main conclusions are summarized as follows:

(1) For the experimental data of multi-period vehicles with few working conditions, the data can be augmented by slicing each period. If the data lengths are inconsistent, the prediction results of data padding by zeros have errors at the end of the sequences, and the model overfits after 12,000 epochs. The prediction results of data supplemented by the LSTM method can fix the original dataset well.

(2) The stability and accuracy of using the GAN models to predict the experimental hydrodynamic data are better than using the transposed convolutional or full connection neural networks alone. As for the selection of hyperparameters of GAN models, the best hidden layer nodes of G1 and D1 are (10, 20, 100, 300, 500) and (100, 80, 50, 20, 1), respectively. The best kernel size of G2 and D2 are three. Among all the GAN models in

this study, the combination of transposed convolutional generator G2 and convolutional discriminator D2 has the best training efficiency and convergence.

(3) The GAN model combined with G2 and D2 performs well in predicting both smoothed time series and mutated data. The mean absolute errors for predicting these two types of data are 0.0434 and 1.4769, which are smaller than the respective experimental data errors of 0.0465 and 1.7249. It is verified that the prediction results of the hydrodynamic time-series data for the manta ray robot by the GAN model are credible.

For the research of hydrodynamic prediction of bionic manta ray underwater vehicles, the data augmentation method of slice data with padding by the LSTM algorithm and the GAN model with G2 + D2 are verified to be accurate and effective. This study will save experimental time and cost and provide detailed hydrodynamic experimental data for the precise control of manta rays.

Author Contributions: Conceptualization, J.B., Q.H. and G.P.; methodology, J.B.; software, J.B.; validation, J.B., G.P., G.P. and J.H.; data curation, J.H.; writing—original draft preparation, J.B.; writing—review and editing, Q.H. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 51879220 and 52001260), National Key Research and Development Program of China (Grant No. 2020YFB1313201), and Fundamental Research Funds for the Central Universities (Grant No. 3102019HHZY030019 and 3102020HHZY030018).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in GitHub URL (<https://github.com/lingke12138/prediction-of-experimental-data.git>) (accessed on 8 September 2022).

Acknowledgments: All individuals included in this section have consented to the acknowledgment.

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

Appendix A

Appendix A.1. The Selection of Hyperparameters in G1 and D1

Here we briefly discuss the number of hidden layers and the number of nodes in each layer of the fully connection neural network generator G1 and discriminator D1 in the GAN model.

For discussing the number of hidden layers, the first and last layers of G1 are set to have 10 and 500 nodes, respectively, and the rest of the layers have 100 nodes. The first layer of D1 has 100 nodes, the last layer has one node, and the others have 10 nodes. We set the number of layers to 3–6, respectively. The hidden layer nodes of different fully connection neural network generators and discriminators are shown in Table A1.

Table A1. Hidden layer nodes of G1 and D1 selected in condition of different layer numbers.

Number of Layers	G1	D1
3	(10, 100, 500)	(100, 10, 1)
4	(10, 100, 100, 500)	(100, 10, 10, 1)
5	(10, 100, 100, 100, 500)	(100, 10, 10, 10, 1)
6	(10, 100, 100, 100, 100, 500)	(100, 10, 10, 10, 10, 1)

The loss function of each GAN model is shown in Figure A1. It can be seen that layer numbers of 5 and 6 have better convergence. To reduce the trainable variables, we select the layer number of 5.

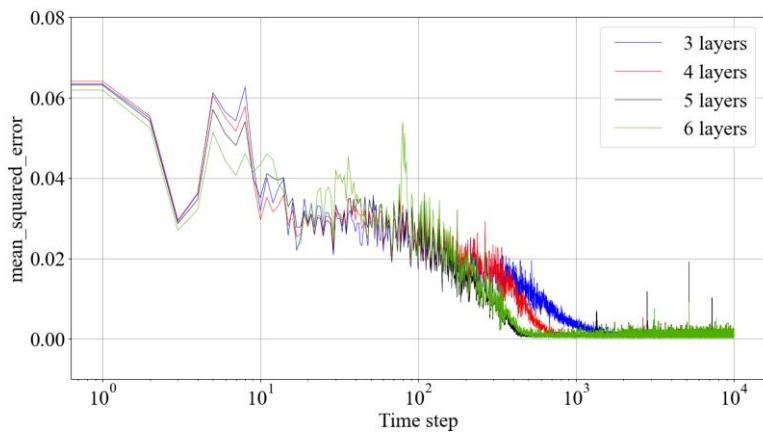


Figure A1. The mean-squared-error validation loss of each G1 + D1 GAN model with the different number of hidden layers.

To discuss the hidden layer nodes, we train the five-layer generator and discriminator model with different node numbers. The number of nodes is shown in Table A2.

Table A2. Hidden layer nodes of G1 and D1 selected in the condition of different layer nodes numbers.

Symbol	G1	D1
GAN_1	(10, 100, 100, 100, 500)	(100, 10, 10, 10, 10, 1)
GAN_2	(10, 20, 100, 300, 500)	(100, 80, 50, 20, 1)
GAN_3	(10, 150, 250, 350, 500)	(100, 75, 50, 25, 1)

The loss values of each GAN model are shown in Figure A2. It can be seen that the GAN_2 model performs better. So the hidden layer nodes of G1 and D1 model selected to Section 3 is (10, 20, 100, 300, 500) and (100, 80, 50, 20, 1) respectively.

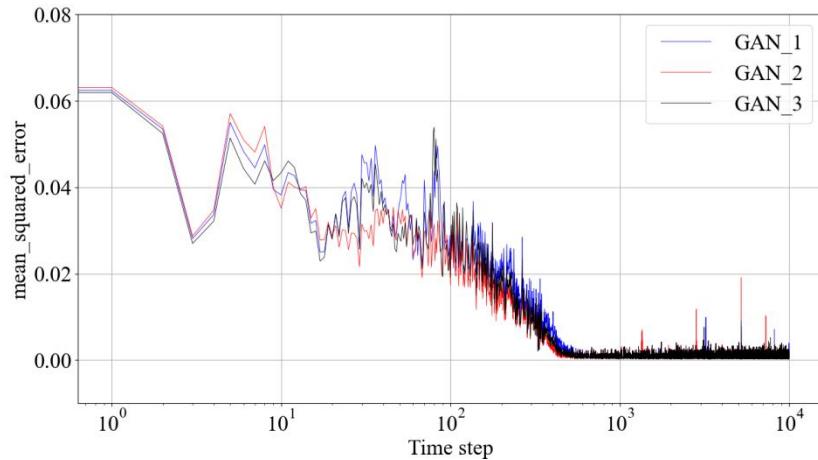


Figure A2. The mean-squared-error validation loss of each G1 + D1 GAN model with different hidden layer nodes.

Appendix A.2. The Selection of Hyperparameters in G2 and D2

G2 and D2 are the transposed convolutional generator and the convolutional discriminator, respectively. In order to convolve the data to the desired data dimension, the stride of G2 is set to (2, 2, 5) for each layer, and the stride of each layer in D2 is set to 4. Based on this, we discuss the performance of different kernel sizes on the prediction results. The loss values trained by GAN models with a kernel size of 3, 5, and 7 are shown in Figure A3. We choose the kernel size of 3 for subsequent predictions on experimental data because it performs more stable.

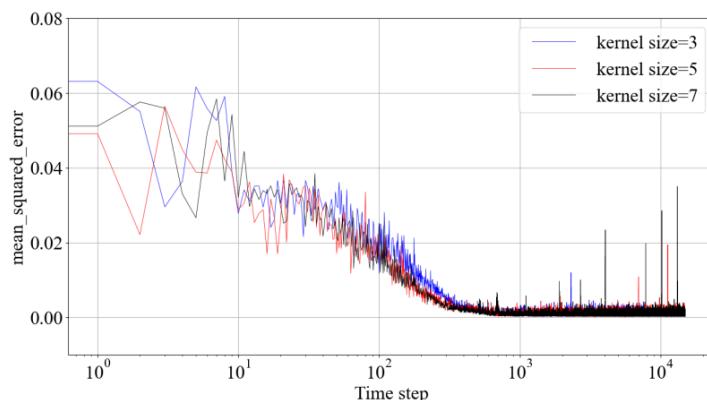


Figure A3. The mean-squared-error validation loss of each G2 + D2 GAN model with different kernel size.

Appendix A.3. The Selection of Learning Rate

As in the previous research in Sections 3.2 and 4.2, we select the G2 + D2 GAN for subsequent predictions on experimental data. We train the model with the Adam optimizer, and the selection of the initial learning rate is discussed here. The validation loss of learning rate equal to 0.01, 0.005, 0.001, and 0.0001 is shown in Figure A4. The model trained by a learning rate of 0.01 and 0.005 is unstable, and 0.0001 is too slow. So the initial learning rate is set to 0.001 for further prediction.

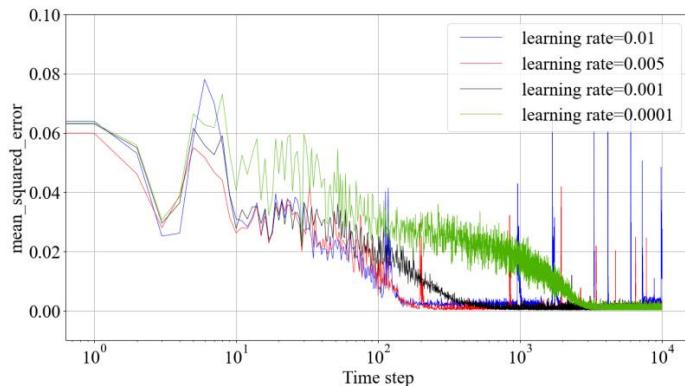


Figure A4. The mean-squared-error validation loss of different initial learning rate.

References

- Li, J.; Bi, S.S.; Gao, J.; Zheng, L.C. Development and Hydrodynamics Experiments of Robotic Manta Ray BH-RAY3. *Control Eng. China* **2010**, *17*, 127–130. [[CrossRef](#)]
- Low, K.H.; Zhou, C.; Seet, G.; Bi, S.; Cai, Y. Improvement and testing of a robotic manta ray (RoMan-III). In Proceedings of the IEEE International Conference on Robotics & Biomimetics, Guangzhou, China, 11–14 December 2012; pp. 1730–1735.
- Feng, D.; Yang, W.; Zhang, Z.; Wang, X.; Yao, C. Numerical study on hydrodynamic behavior of flexible multi-stage propulsion foil. *AIP Adv.* **2021**, *11*, 035326. [[CrossRef](#)]
- Lindsey, C.C. Form, Function and Locomotory Habits in Fish. In *Fish Physiology VII*; Hoar, W.S., Randall, D.J., Eds.; Academic Press: Cambridge, MA, USA, 1978; pp. 1–100.
- Fish, F.E.; Kolpas, A.; Crossett, A.; Dudas, M.A.; Moored, K.W.; Bart-Smith, H. Kinematics of swimming of the manta ray: Three-dimensional analysis of open-water maneuverability. *J. Exp. Biol.* **2018**, *221*, jeb.166041. [[CrossRef](#)] [[PubMed](#)]
- Clark, R.; Smits, A. Visualizations of the Unsteady Wake of Manta Ray Model. In Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 9–12 January 2006.
- Bottom, R. On the hydrodynamics of ray-like swimming. In *Meeting of the Aps Division of Fluid Dynamics*; American Physical Society: College Park, MD, USA, 2014.
- Cai, Y.; Bi, S.; Zheng, L. Design and experiments of robotic fish imitating cow-nosed ray. *J. Bionic Eng.* **2010**, *7*, 120–126. [[CrossRef](#)]
- Zhou, C.; Low, K.-H. Better Endurance and Load Capacity: An Improved Design of Manta Ray Robot (RoMan-II). *J. Bionic Eng.* **2010**, *7*, S137–S144. [[CrossRef](#)]
- Fish, F.E.; Schreiber, C.M.; Moored, K.M.; Liu, G.; Dong, H.; Bart-Smith, H. Hydrodynamic Performance of Aquatic Flapping: Efficiency of Underwater Flight in the Manta. *Aerospace* **2016**, *3*, 20. [[CrossRef](#)]

11. Huang, H.; Sheng, C.; Wu, J.; Wu, G.; Zhou, C.; Wang, H. Hydrodynamic analysis and motion simulation of fin and propeller driven manta ray robot. *Appl. Ocean Res.* **2021**, *108*, 102528. [[CrossRef](#)]
12. Yang, S.; Jing, Q.; Han, X. Kinematics Modeling and Experiments of Pectoral Oscillation Propulsion Robotic Fish. *J. Bionic Eng.* **2009**, *6*, 174–179. [[CrossRef](#)]
13. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *Comput. Sci.* **2015**. [[CrossRef](#)]
14. Dollar, P.; Tu, Z.; Belongie, S. Supervised Learning of Edges and Object Boundaries. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 1964–1971. [[CrossRef](#)]
15. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017.
16. Leordeanu, M.; Sukthankar, R.; Hebert, M. Unsupervised Learning for Graph Matching. *Int. J. Comput. Vis.* **2012**, *96*, 28–45. [[CrossRef](#)]
17. Hussain, I.; Park, S.J. HealthSOS: Real-Time Health Monitoring System for Stroke Prognostics. *IEEE Access* **2020**, *88*, 213574–213586. [[CrossRef](#)]
18. Hussain, I.; Park, S.J. Big-ECG: Cardiographic Predictive Cyber-Physical System for Stroke Management. *IEEE Access* **2021**, *9*, 123146–123164. [[CrossRef](#)]
19. Dong, W.; Li, S.; Fu, X.; Li, Z.; Fairbank, M.; Gao, Y. Control of a Buck DC/DC Converter Using Approximate Dynamic Programming and Artificial Neural Networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 1760–1768. [[CrossRef](#)]
20. Wang, Z.; Xiao, D.; Fang, F.; Govindan, R.; Pain, C.; Guo, Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int. J. Numer. Methods Fluids* **2017**, *86*, 265–268. [[CrossRef](#)]
21. Wu, Z.; Fan, D.; Zhou, Y.; Li, R.; Noack, B.R. Jet mixing optimization using machine learning control. *Exp. Fluids* **2018**, *59*, 131. [[CrossRef](#)]
22. Lee, Y.; Yang, H.; Yin, Z. PIV-DCNN: Cascaded deep convolutional neural networks for particle image velocimetry. *Exp. Fluids* **2017**, *58*, 171. [[CrossRef](#)]
23. Mohan, A.T.; Gaitonde, D.V. A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks. *arXiv* **2018**, arXiv:1509.02971.
24. Lopez-Martin, M.; Clainche, S.L.; Carro, B. Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network. *Expert Syst. Appl.* **2021**, *177*, 114924. [[CrossRef](#)]
25. Thavarajah, R.; Zhai, X.; Ma, Z.; Castineira, D. Fast Modeling and Understanding Fluid Dynamics Systems with Encoder-Decoder Networks. *Mach. Learn. Sci. Technol.* **2020**, *2*, 025022. [[CrossRef](#)]
26. Li, J.; Zhang, M.; Martins, J.R.R.A.; Shu, C. Efficient Aerodynamic Shape Optimization with Deep-Learning-Based Geometric Filtering. *AIAA J.* **2020**, *58*, 1–17. [[CrossRef](#)]
27. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco, CA, USA, 13–17 August 2016.
28. Murata, T.; Kai, F.; Fukagata, K. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.* **2020**, *882*, A13. [[CrossRef](#)]
29. Laubscher, R. Simulation of multi-species flow and heat transfer using physics-informed neural networks. *arXiv* **2021**, arXiv:2105.14907. [[CrossRef](#)]
30. Rao, C.; Sun, H.; Liu, Y. Physics-informed deep learning for incompressible laminar flows. *Theor. Appl. Mech. Lett.* **2020**, *10*, 207–212. [[CrossRef](#)]
31. Wang, H.; Yang, Z.; Li, B.; Wang, S. Predicting the near-wall velocity of wall turbulence using a neural network for particle image velocimetry. *Phys. Fluids* **2020**, *32*, 115105. [[CrossRef](#)]
32. Cai, S.; Wang, Z.; Fuest, F.; Jeon, Y.J.; Gray, C.; Karniadakis, G.E. Flow over an espresso cup: Inferring 3D velocity and pressure fields from tomographic background oriented schlieren videos via physics-informed neural networks. *J. Fluid Mech.* **2021**, *915*. [[CrossRef](#)]
33. Panda, J.P.; Warrior, H. Data-driven prediction of complex flow field over an axisymmetric body of revolution using Machine Learning. *J. Offshore Mech. Arct. Eng.* **2022**, *8*, 1–30. [[CrossRef](#)]
34. Panda, J.P. Machine Learning for Naval Architecture, Ocean and Marine Engineering. *arXiv* **2021**, arXiv:2109.05574.
35. Yang, Y.; Tu, H.; Song, L.; Chen, L.; Xie, D.; Sun, J. Research on Accurate Prediction of the Container Ship Resistance by RBFNN and Other Machine Learning Algorithms. *J. Mar. Sci. Eng.* **2021**, *9*, 376. [[CrossRef](#)]
36. Grabe, C.; Knopp, T.; Eisfeld, B.; Jackel, F. Towards the Border of the Flight Envelope: Strategies to improve RANS Turbulence Models. In Proceedings of the Kolloquium für Mechanik, Brunswick, Germany, 14 January 2021.
37. Thuerey, N.; Weissenow, K.; Prantl, L.; Hu, X. Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows. *AIAA J.* **2020**, *58*, 25–36. [[CrossRef](#)]
38. Kalsi, H.S.; Tucker, P.G. Numerical Modelling of Shock Wave Boundary Layer Interactions in Aero-engine Intakes at Incidence. In Proceedings of the ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, Oslo, Norway, 11–15 June 2018.