

## Article

# Compressive Sensing Convolution Improves Long Short-Term Memory for Ocean Wave Spatiotemporal Prediction

Lingxiao Zhao <sup>1,2</sup> , Yijia Kuang <sup>2</sup> , Junsheng Zhang <sup>1,\*</sup> and Bin Teng <sup>3</sup><sup>1</sup> College of Ocean and Civil Engineering, Dalian Ocean University, Dalian 116023, China; lxzhao23@m.fudan.edu.cn<sup>2</sup> Department of Atmospheric and Oceanic Sciences, Fudan University, Shanghai 200438, China; 23213020011@m.fudan.edu.cn<sup>3</sup> State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China; bteng@dlut.edu.cn

\* Correspondence: yzjz009@hotmail.com

## Abstract

This study proposes a Compressive Sensing Convolutional Long Short-Term Memory (CSCL) model that aims to improve short-term (12–24 h) forecast accuracy compared to standard ConvLSTM. It is especially useful when subtle spatiotemporal variations complicate feature extraction. CSCL uses uniform sampling to partially mask spatiotemporal wave fields. The model training strategy integrates both complete and masked samples from pre- and post-sampling. This design encourages the network to learn and amplify subtle distributional differences. Consequently, small variations in convolutional responses become more informative for feature extraction. We considered the theoretical explanations for why this sampling-augmented training enhances sensitivity to minor signals and validated the approach experimentally. For the region 120–140° E and 20–40° N, a four-layer CSCL model using the first five moments as inputs achieved the best prediction performance. Compared to ConvLSTM, the  $R^2$  for significant wave height improved by 2.2–43.8% and for mean wave period by 3.7–22.3%. A wave-energy case study confirmed the model's practicality. CSCL may be extended to the prediction of extreme events (e.g., typhoons, tsunamis) and other oceanic variables such as wind, sea-surface pressure, and temperature.

**Keywords:** compressive sensing convolution; long short-term memory; spatiotemporal; significant wave height; mean wave period



Academic Editor: Liliana Rusu

Received: 22 July 2025

Revised: 28 August 2025

Accepted: 2 September 2025

Published: 4 September 2025

**Citation:** Zhao, L.; Kuang, Y.; Zhang, J.; Teng, B. Compressive Sensing Convolution Improves Long Short-Term Memory for Ocean Wave Spatiotemporal Prediction. *J. Mar. Sci. Eng.* **2025**, *13*, 1712. <https://doi.org/10.3390/jmse13091712>

**Copyright:** © 2025 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

Offshore renewable energy facilities, such as Wave Energy Converters (WECs), have rapidly expanded in recent years to capture the enormous energy in waves [1]. In the advancement of WEC technology, wave prediction has received much attention as a key driver of energy harvesting efficiency [2]. It has been applied in various fields including observatory monitoring [3], collection efficiency assessment [4], and safety detection under extreme wave conditions [5]. Due to the intermittent and random nature of waves, accurate prediction of future or historical wave states can both promote the continuity of harvesting operations along with wave farm optimization [6]. Wave forecasting techniques benefit the entire offshore industry in, for example, evaluating the exposure of offshore structures such as wind turbines or bridges to extreme wave loads, or the optimization of daily

routes [7–9]. This study focuses on the prediction of wave height and period—factors that directly determine the accuracy of wave energy prediction.

Although field measurements allow for direct and reliable recording of wave height, the high cost and specialized nature of the equipment limit the data collection to few areas [10]. Numerical wave simulation has become an important tool to address the lack of real-world data. Mainstream tools including WAVEWATCH and Simulating Waves Nearshore (SWAN) rely on the third-generation wave simulation framework [11,12]. The numerical simulations reflect the physical mechanisms of wave generation and propagation by inputting spatiotemporal wind field data associated with boundary wave conditions to capture relevant wave information [13]. For example, Shankar, Cambazoglu, Bernstein, Hesser, and Wiggert [14] used WAVEWATCH III to develop a multigrid nested model for assessing the importance of atmospheric forcing on wind wave modeling for extreme hurricane conditions in the Gulf of Mexico basin. However, numerical simulations are computationally demanding in large-scale applications, limiting real-time output and retarding the evaluation of long-term large-scale historical wave height data [15].

The principle of wave spatiotemporal forecasting involves establishing the underlying spatiotemporal relationships between input data and wave fields [16]. Mainstream wave forecasting methods primarily employ deterministic forecasting [17]. Since waves are a meteorological phenomenon, wave height forecasting has also been influenced by advancements in weather forecasting. Some empirical statistical methods have provided new insights and greatly advanced the field of wave height forecasting, such as the analogy method [18], autoregression [19], autoregressive integrated moving average (ARIMA) [20], and principal component analysis [21].

Due to the rapid development of artificial intelligence, machine learning methods have achieved great success in various disciplines due to their powerful capabilities in implicit nonlinear regression [22]. In wave height prediction, machine learning methods have been employed in deterministic wave height forecasting [23–25]. Zhang, Li, Gao, and Ren [17] showed that using both observed data and SWAN results as inputs in long short-term memory (LSTM) greatly improved temporal prediction accuracy compared to SWAN. Building on the work of Zhang, Li, Gao, and Ren [17], Li, Zhang, Lyu, and Zhang [26] added the self-attention mechanism. Some well-established work uses neural networks and historical wave data for future wave height prediction [27,28]. Yao and Wu [29] used wind speed and wave height to establish a one-step-ahead wave height prediction method, while Wang and Ying [30] utilized kernel density estimation (KDE) to determine the probability density distribution of prediction errors generated from wave height prediction intervals.

Two main conceptual approaches exist in wave forecasting (Table 1). The first is physics-based prediction [31]. This approach enhances the model by explicitly incorporating domain knowledge during the learning process, such as historical meteorological factors physically linked to wave evolution, or physical constraints such as conservation laws, thereby improving the physical consistency and generalizability of the forecast results. The second is heuristic-based prediction, which uses historical wave time series to predict future conditions [32]. This method does not introduce exogenous meteorological covariates but instead leverages the temporal dependency of the target variable itself. Modern derivatives include simple autoregression and ARIMA models, as well as modern sequence learners trained solely on historical wave observations (e.g., LSTM and ConvLSTM) [33]. Since no additional inputs are required, these methods have lower computational overhead and are easier to engineer and deploy. This study focuses on the heuristic-based approach. Both types of approaches have received largely equal attention in the field and have been accepted in the mainstream. Physics-based prediction rarely optimizes the model structure itself but instead trains the model using a large amount of auxiliary data (e.g., UV wind,

sea surface pressure, sea surface temperature) to improve prediction accuracy [34]. Current applications are limited by data availability, which inhibits real-time forecasting [35]. In heuristic-based prediction, when using only historical wave data, the research has focused mainly on improving model structure [36], though these studies are more challenging.

**Table 1.** Summary of research on wave forecasting.

| Ref.                                                  | Model                                              | Duration | Type      | Dimensions   |
|-------------------------------------------------------|----------------------------------------------------|----------|-----------|--------------|
| Ti and Kong [37]                                      | Generative adversarial network                     | 3 h      | Heuristic | Spatial (2D) |
| Zhang, Luo, Quan, Wang, Shi, Shen and Zhang [38]      | Convolutional long short-term memory               | 3 h      | Heuristic | Time (1D)    |
| Liu, Lu, Wang, Lai, Ying, Li, Han, Wang and Dong [39] | Earthformer                                        | 1 h      | Physics   | Spatial (2D) |
| Wang, Bethel, Xie and Dong [40]                       | Empirical wavelet transform–long short-term memory | 6 h      | Heuristic | Time (1D)    |
| Jörges, Berken-brink, Gottschalk and Stumpe [41]      | Convolutional neural network                       | 1 h      | Physics   | Spatial (2D) |
| Yevnin, Chorev, Dukan and Tole-do [42]                | Gated recurrent unit                               | 1 h      | Heuristic | Spatial (2D) |
| Meng, Xu and Song [43]                                | Adaptive time–frequency neural network             | 1 h      | Heuristic | Time (1D)    |
| Dixit, Londhe and Dandawate [44]                      | Neuro-wavelet modeling                             | 6 h      | Heuristic | Time (1D)    |
| Zilong, Yubing and Xiaowei [45]                       | Convolutional long short-term memory               | 3 h      | Physics   | Spatial (2D) |
| Ouyang, Zhao, Zhang and Zhang [46]                    | Convolutional neural network                       | 1 h      | Heuristic | Spatial (2D) |
| Luo, Shi, Zhang, Zhang, Zhou, Pan and Wang [47]       | Decision trees and multi-layer perceptron          | 1 h      | Physics   | Spatial (2D) |

Although ConvLSTM has been widely used in spatiotemporal deterministic series forecasting tasks, no adapted module has been developed for longer lead time forecasting, thus the model is only useful for 0–12 h nowcasting. For short-term forecasts of 12–24 h, the accuracy of ConvLSTM drops below the acceptable threshold [30,38,45]. Some studies have attempted to address these issues through probabilistic forecasting. Unlike direct prediction of a single numerical value, probabilistic forecasting aims to learn the predictive distribution or probability density of the target variable, effectively representing forecast uncertainty and providing more reliable confidence information [48]. Such methods (e.g., generative adversarial networks [49], denoising diffusion probabilistic models [50]) have demonstrated superior uncertainty quantification capabilities and more robust performance in both short-term and medium-to-long-term forecasts (12–24 h and beyond). In particular, Price et al.’s GenCast outperforms ensemble forecast in terms of spread/skill for lead times of 1–15 days.

Despite the widespread attention given to probabilistic forecasting, research in the field of wave prediction is limited. Additionally, probabilistic methods typically require more computational resources and engineering effort: for example, Price et al. conducted model training and parameter tuning on 52 TPU-v5 units, which exceeds the equipment capabilities of most operational institutions and ordinary engineering resources. Therefore, the continued development of deterministic forecasts, which are structurally simple and require lower operational conditions, remains highly important. Furthermore, the numerous wave forecasting studies shown in Table 1 are all based on deterministic forecasting frameworks, highlighting the direct importance of deterministic models in existing operational workflows. Accordingly, this study focused on improving the deterministic forecasting framework of ConvLSTM, drawing inspiration from certain modeling concepts in probabilistic forecasting rather than directly adopting a full probabilistic modeling approach.

This study theoretically and experimentally investigated the performance of the Compressive Sensing Convolution Long Short-Term Memory (CSCL) model for learning feature representations under partial information conditions and inferring complete information. We hypothesized that, if the model is forced to extract key features from incomplete images by artificially masking part of the spatial pixels during training, regression accuracy can be increased by virtue of complete inference and stronger feature extraction ability. This principle is demonstrated in text processing using the masked language model [51]. Contextual information is learned by randomly masking some words in a sentence and training the model to predict the masked words [52]. We incorporated some facets of compressive

sensing, which can recover complete information using partial information under many assumptions in the digital signaling domain [53].

Fully lossless compressive sensing currently exists only in 1D time series and is constrained by data sparsity and uncertainty, which limits its application in spatiotemporal series data. One solution may be using simple uniform sampling and adding both pre- and post-sampling data to model training. Since uniform sampling introduces small losses in finite data [54], CSCL is trained using complete information pre-sampling and information from masked spatial pixels in post-sampling. We evaluated the small difference in the amount of information between the pre- and post-sampling data using KDE and Kullback–Leibler (KL) divergence and found that the CSC operation was able to amplify these small differences in training, enabling sensitive model prediction. Further theoretical derivation proved the statistical results. In this study, the experiments were conducted on spatiotemporal series data with 1 h time intervals and  $0.5^\circ$  spatial resolution over the region  $120\text{--}140^\circ$  E and  $20\text{--}40^\circ$  N, and the significant wave height (SWH) and mean wave period (MWP) were trained and validated, respectively. Finally, the annual and seasonal mean wave energy in the region were calculated and evaluated using the model.

## 2. Methods

This study employed 5-year spatiotemporal series data. The extraction and learning abilities of CSC were used for spatial features. The temporal features of the data were extracted using the long-term memory ability of LSTM.

### 2.1. Statistical Assessment

#### 2.1.1. Kernel Density Estimation

KDE is a nonparametric method that applies kernel smoothing to probability density estimation, i.e., using kernel weight to estimate the probability density function of a random variable [55]. KDE can infer the total population based on a limited number of data samples and effectively represents the probability distribution of the data.

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right), \quad (1)$$

where  $n$  is the number of samples,  $h$  is the smoothing parameter (bandwidth) controlling the width of the kernel function,  $K(\cdot)$  is the kernel function, and  $x_i$  is the sample point.

#### 2.1.2. Kullback–Leibler Divergence

KL divergence is used to measure the loss of information between two probability distributions [56]:

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx, \quad (2)$$

where  $P(x)$  represents the distribution of the original data and  $Q(x)$  represents the estimated distribution at different sampling rates.

### 2.2. Long Short-Term Memory

LSTM is a recurrent neural network (RNN) [57] that utilizes a recursive mechanism and gate technique to overcome gradient explosion and gradient disappearance [58] to extract dependencies from information sequences. The biggest advantage of LSTM over RNN is its ability to remember long-term information and forget unimportant information [59].

The LSTM comprises several recurrent cells with inputs containing the current moment data  $x_t$ , previous cell state vector  $C_{t-1}$ , and previous hidden layer output vector  $h_{t-1}$

(Figure 1). The LSTM first computes the cell's discard information  $f_t$  through a forget gate, with a size of  $[0, 1]$  (the smaller the value, the higher the degree of forgetting).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

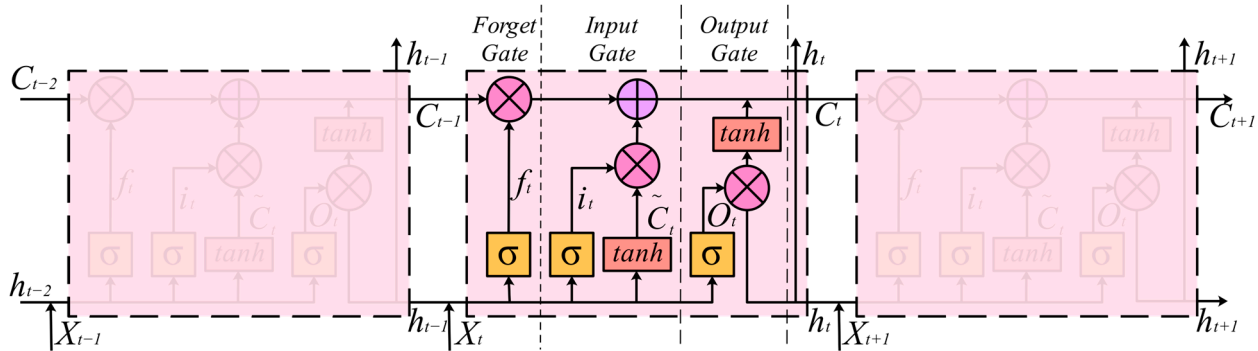


**Figure 1.** The LSTM model structure.

Here,  $f_t$  is the output of the forget gate,  $W_f$  is the weight matrix of the forget gate,  $[h_{t-1}, x_t]$  represents the connection of the two vectors,  $b_f$  is the offset term of the forget gate, and  $\sigma$  represents the sigmoid activation function  $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1+\tanh(\frac{z}{2})}{2}$ .

The input gate receives the new candidate cell  $\tilde{C}_t$  via  $x_t$  and  $h_{t-1}$  and refines and updates  $C_{t-1}$ .

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (5)$$

Here,  $i_t$  is the output of the input gate;  $b_i$  and  $b_C$  are the offset terms of the input gate and candidate cell, respectively;  $W_i$  and  $W_C$  are the weight matrices of the input gate and candidate cell, respectively; and  $\tanh$  is the activation function  $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ .

The cell state vector  $C_t$  at the current moment is represented as

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t. \quad (6)$$

The final output gate uses  $\sigma$  to determine the cell state and the output  $h_t$  is obtained by  $\tanh$ .

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (8)$$

Here,  $o_t$  is the result of the output gate and  $b_o$  is the offset term of the output gate.

### 2.3. Compressive Sensing Convolution LSTM

In traditional convolutional neural networks, the convolutional kernel performs computations on the input data to learn local features. However, this approach may not effectively capture the small structural changes that are critical to the prediction task. To enhance the network's ability to perceive critical details, we introduced CSC, enabling effective feature extraction.

#### 2.3.1. Compressed Sensing

Compressed sensing (CS) theory [60] states that, if a signal is sparse in a certain transform domain, then the Nyquist sampling theorem dictates that the sampling frequency must be two-fold higher than the highest signal frequency to retain the complete informa-

tion of the original signal. Assuming that the original signal  $x$  can be sparsely represented on some base  $\Psi$ ,  $x = \Psi\alpha$ , then

$$y = \Phi x = \Phi \Psi \alpha, \quad (9)$$

where  $\alpha$  is a sparse coefficient vector and  $\Phi$  is the sampling matrix. CS essentially functions by using the measurement matrix  $\Phi$  for downsampling. As long as  $\Phi$  satisfies the restricted isometry property [61],  $x$  can be recovered by the optimization method.

However, two main challenges limit the application of CS in spatiotemporal series prediction: (i) CS is mainly used for time-series signals, while its applicability in spatiotemporal series analysis is unclear [62]. (ii) A suitable sampling strategy is required to ensure the full representation of information in the original data [63].

With sufficiently long time-series records, the data distribution and information content before and after uniform sampling are essentially the same [64]. However, with limited data, uniform sampling causes a small loss of information. Since CS is not appropriate for spatiotemporal data, we utilized the post-sampling data for training and gradually introduced the pre-sampling data, thus allowing the model to learn the effects of the small differences between pre- and post-sampling and improving the prediction performance.

### 2.3.2. Objective Optimization Function

The CSCL model output  $f_\theta(\cdot)$  should be as close as possible to the true value.

$$\min_{\theta} L(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\|f_\theta(x) - y\|^2] \quad (10)$$

Here,  $\theta$  is the trainable parameter,  $L(\theta)$  denotes the training loss,  $x$  is the input sample,  $y$  is the true value, and  $\mathbb{E}$  is taken from all the samples in the data distribution  $\mathcal{D}$ .

Since different spatial sampling rates introduce small information differences, the model sensitivity to the input perturbation  $\delta x$  must be enhanced to better learn the local features. Based on the first-order approximation, we obtain

$$\delta f(x) = f_\theta(x + \delta x) - f_\theta(x) \approx J_{f_\theta}(x) \delta x, \quad (11)$$

where  $J_{f_\theta}(x) = \frac{\partial f_\theta}{\partial x}$  is the Jacobian matrix of the CSCL model at  $x$  [65]. To make the model more sensitive to perturbations in the critical direction, we introduced an auxiliary target term to amplify local perturbations in the critical direction during learning:

$$\min_{\theta} L(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\|f_\theta(x) - y\|^2] - \lambda \mathbb{E}_{x \sim \mathcal{D}} [\log \|J_{f_\theta}(x)\|], \quad (12)$$

where  $\lambda > 0$  is a trade-off factor between the loss term and perturbation sensitivity. This optimization not only minimizes model prediction error but also improves its sensitivity to local perturbations and ability to capture critical details.

### 2.3.3. Design of Compressive Sensing Convolution

In the CSC framework, the convolutional operation should perform spatial feature extraction and also simulate the CS process, which enhances the discriminative power of the network.

$$f_\theta(x) = W *_r (\Phi x) + b \quad (13)$$

Here,  $x \in \mathbb{R}^{C \times H \times W}$  is the input data,  $\Phi$  is the measurement matrix responsible for sampling the input  $x$ ,  $*_r$  is a uniform sample with sampling rate  $r$ ,  $W$  is a trainable

weight matrix, and  $b$  is a bias term. The weight matrix  $W$  is locally weighted to sum the sampled data:

$$f_{\theta}(x)(p, q) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} W_{ij}(\Phi x)(p + s \cdot i, q + s \cdot j) + b, \quad (14)$$

where  $(p, q)$  is the position in the output feature map,  $k \times k$  is the size of the kernel, and  $s = \frac{1}{r}$ . The CSCL model workflow is shown in Figure 2.

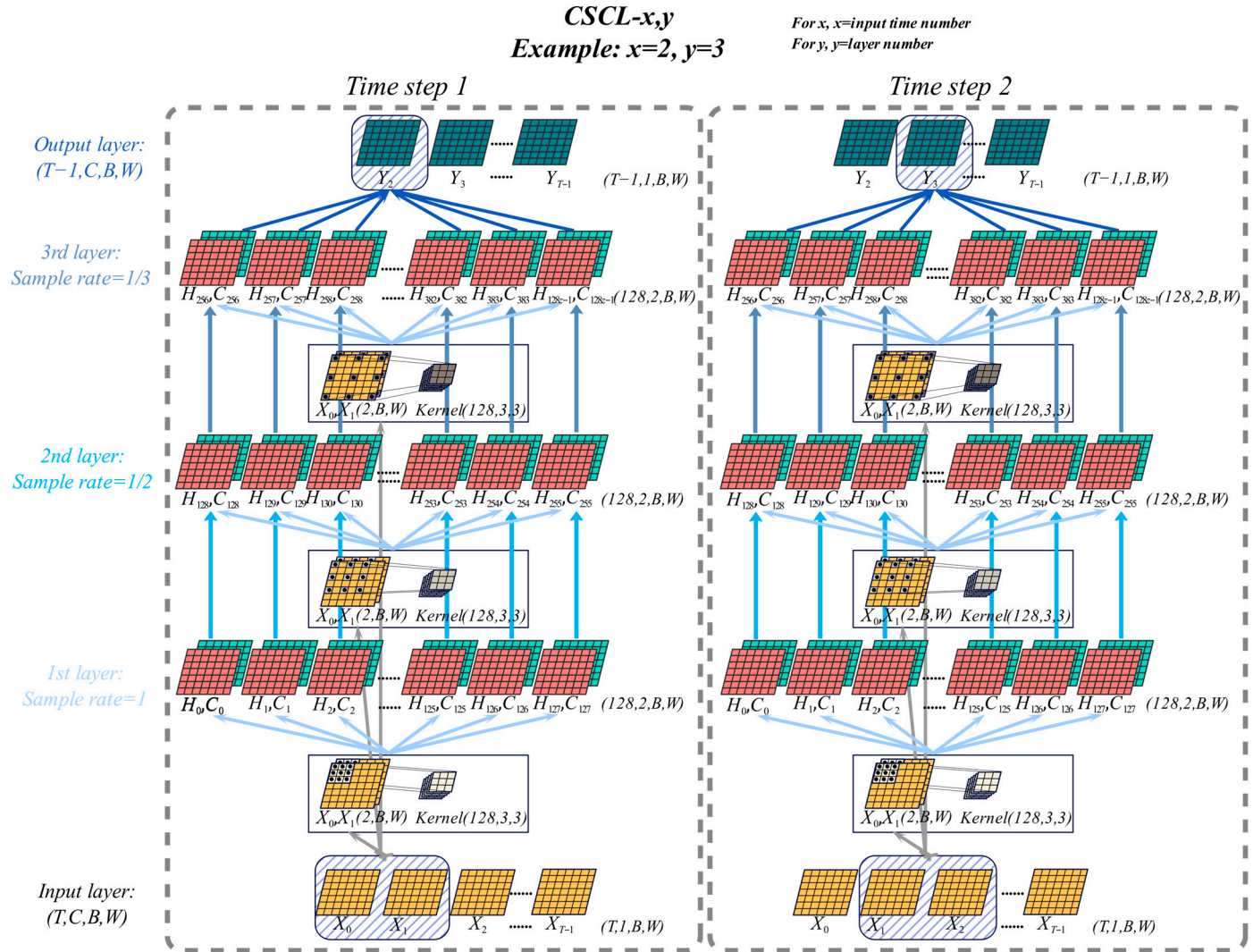


**Figure 2.** CSCL model workflow.

#### 2.4. Error Metrics

To rigorously evaluate the performance of the forecasting model, we adopted three standard metrics: root mean square error (RMSE), mean absolute error (MAE), and the coefficient of determination ( $R^2$ ) [66].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2}, \quad (15)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - x'_i|, \text{ and} \quad (16)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (x_i - x'_i)^2}{\sum_{i=1}^n (x_i - \bar{x}_i)^2}, \quad (17)$$

where  $x_i$  and  $x'_i$  are the true and predicted time  $i$  and  $n$  is the number of test-sets.

### 3. Data Analysis and Theory

#### 3.1. Study Area

The SWH and MWP datasets (covering 1 January 2019–31 December 2023; grid resolution of  $0.5^\circ \times 0.5^\circ$ ) originated from the ERA5 reanalysis produced by the European Centre for Medium-Range Weather Forecasts (available at: <https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels?tab=overview>, accessed on 15 February 2025) [67]. This reanalysis dataset integrates numerical model outputs with global observational data based on physical principles, ensuring a spatially and temporally coherent representation [68]. The study area spans  $120^\circ\text{E}$ – $140^\circ\text{E}$  and  $20^\circ\text{N}$ – $40^\circ\text{N}$  (Figure 3).

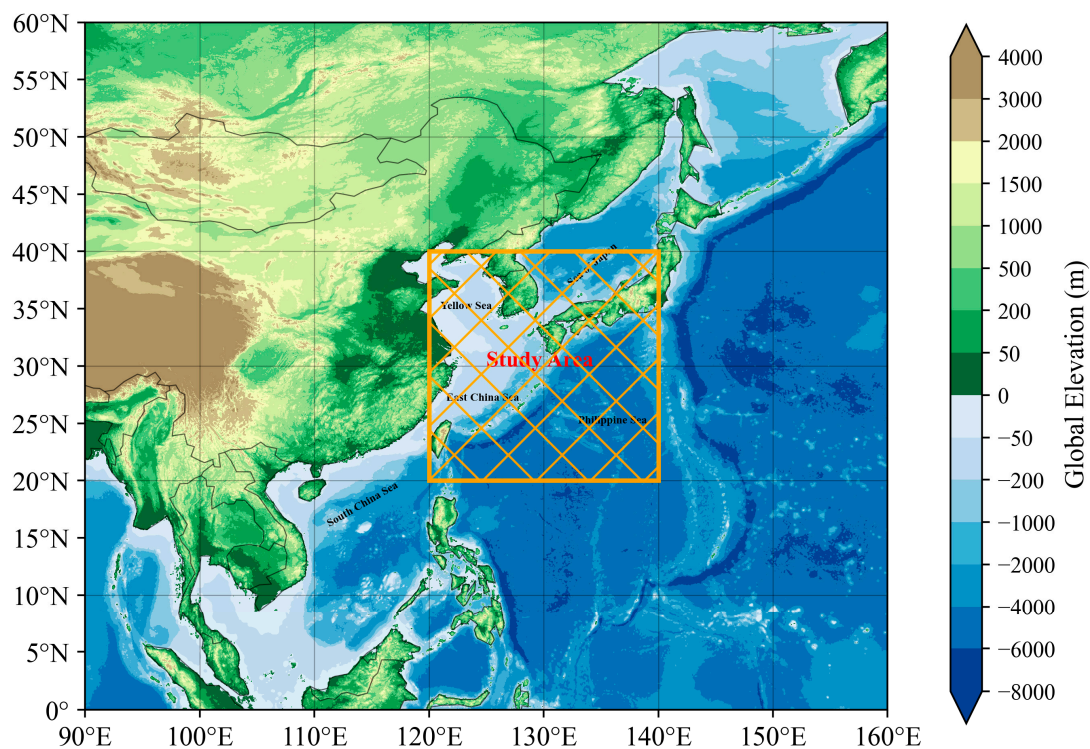


**Figure 3.** Study area.

The dataset was partitioned into three subsets: 70% for training, 10% for validation, and 20% for testing. Specifically, the training set comprised 30,678 samples, validation set comprised 4382 samples, and test-set contained 8764 samples. This data split ensures the model has sufficient samples for learning while maintaining a separate validation set for hyperparameter tuning and test-set for evaluating generalization performance.

#### 3.2. KDE and KL Divergence Under CSC

We compared the difference in distribution between the original data and the data after convolution with a  $3 \times 3$  average filter at different resolutions (Figures 4 and 5).

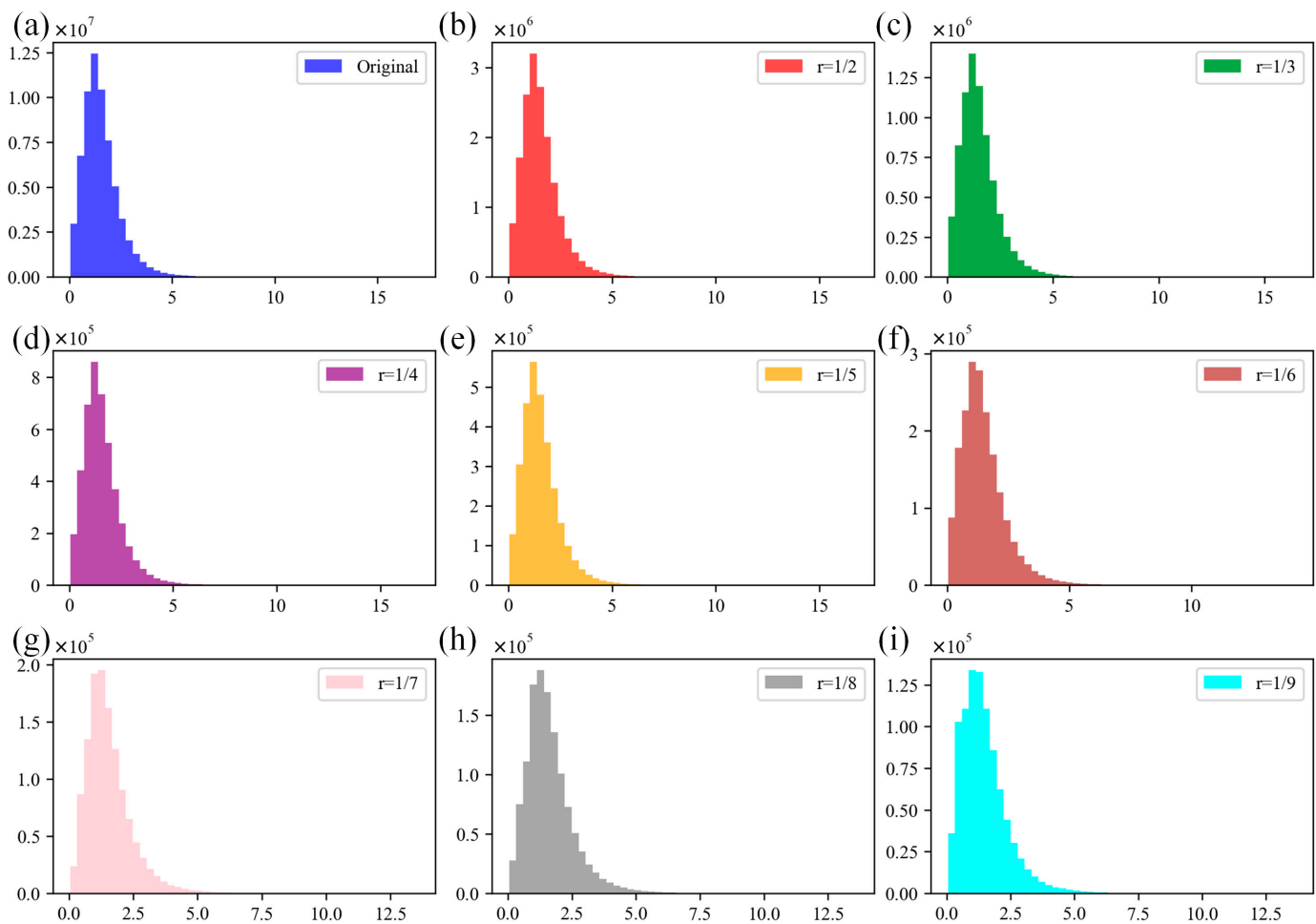


**Figure 4.** Distribution of original data at different sampling rates  $r$ . (a) Original data. (b–i)  $r = 1/2$  to  $1/9$ .

There was no obvious difference in the distribution of the original data (Figure 4b–e), and the overall distribution was still skewed to the left even at  $r = 1/5$  to  $1/9$  (Figure 4e–i). After average filter convolution, the plot in Figure 5i varied significantly from that in Figure 5a. The difference between the probability density distributions was further evaluated using the KDE (Figure 6).

The original data and  $r = 1/2$  to  $1/6$  distributions differed dramatically from the  $r = 1/7$  to  $1/9$  distributions (Figure 6a). A large difference was observed in the distributions from  $r = 1/2$  to  $1/9$  (Figure 6b); their kurtosis was relatively centralized, and the skewness also increased. The specific differences were evaluated based on the KL divergence (Table 2) for the Original and Convolved1 data (Convolved2 to Convolved4 are the results of other types of convolutional kernels).

In the original data, the KL divergences for  $r = 1/2$  to  $1/6$  were all very small, and the KL divergence for  $r = 1/7$  to  $1/9$  increased but was still much smaller than after other kernel convolutions. This indicates high similarity in data distribution at different sampling rates, with only minor differences. However, the convolution operation (Convolved1–Convolved4) dramatically amplified the subtle differences in the distributions of the different sampling rates by ~300-fold (average filter) to >10,000-fold (Laplacian filter). Consequently, we hypothesized that the small differences in the original data are gradually amplified by arbitrary convolutional machine learning computation and, since the model training accounts for these differences, the model accuracy can be substantially improved.

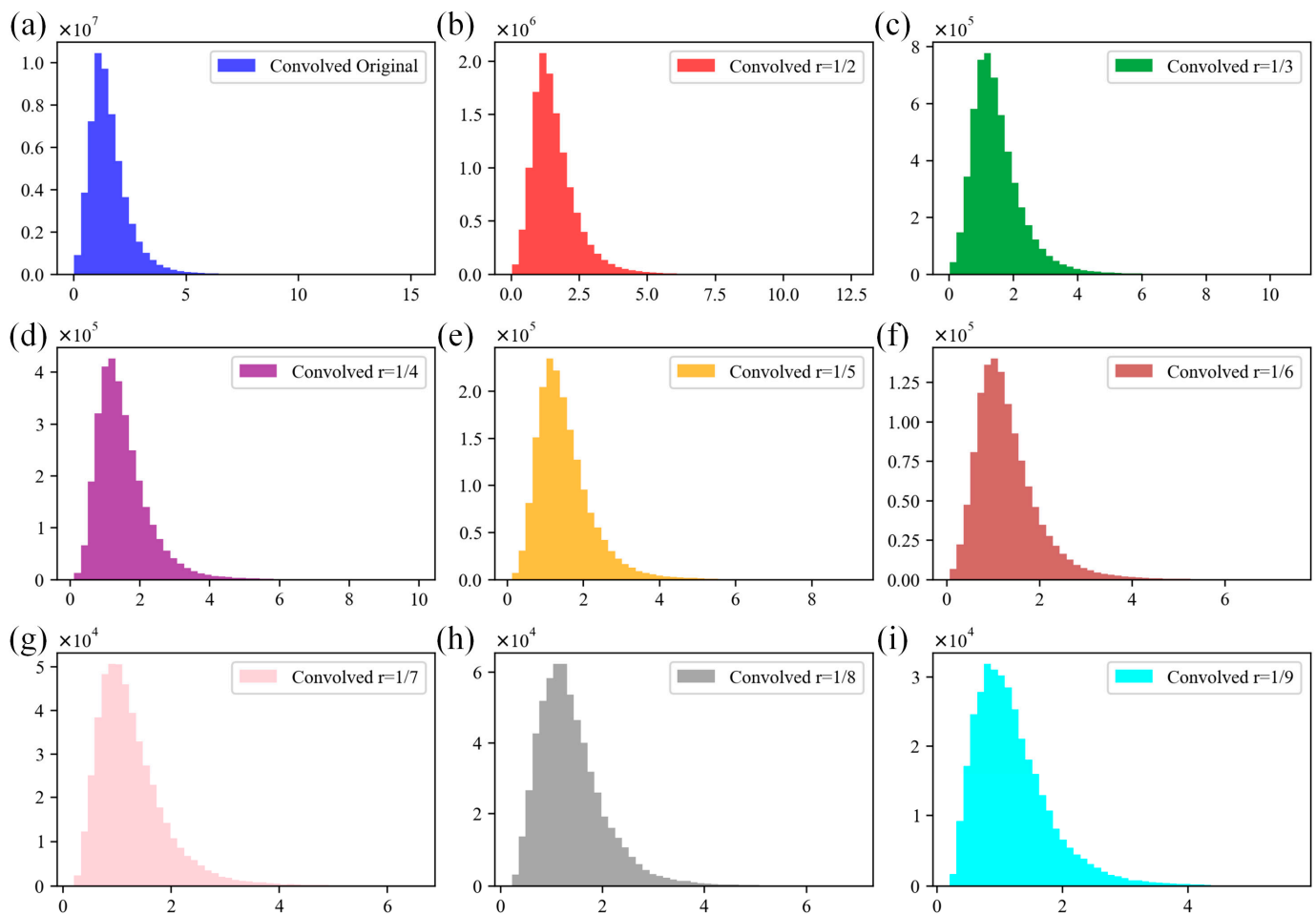


**Figure 5.** Distribution of data after  $3 \times 3$  average filter at different sampling rates  $r$ . (a) Convolved data. (b–i)  $r = 1/2$  to  $1/9$ .

### 3.3. Preservation of Mutual Information Under CSC

This study used both theoretical and experimental approaches to determine why the CSCL architecture outperforms the traditional ConvLSTM in 12–24 h forecast accuracy. First (Section 3.3.1), we proved from an information-theory perspective that, under the reasonable assumptions of steady and uniform sampling, the compressed sampling operation causes minimal information loss ( $D_{KL} \approx 0$ ) for the target distribution, preserving most of the mutual information. Secondly (Sections 3.3.2 and 3.3.3), we demonstrated that convolutional layers amplify task-relevant small local perturbations differentially during training. The cumulative amplification effect, combined with gating/memory mechanisms, enhances the network's sensitivity to critical information. Together, these mechanisms explain why CSCL can extract and amplify weak features with predictive value under sparse sampling conditions, thereby improving the accuracy of longer-term forecasts.

**Table 2.** KL divergence of different kernels and sampling rates.

| Name       | Kernel            | Compressive Sensing Sampling Rate $r$ |         |         |         |         |         |         |         |
|------------|-------------------|---------------------------------------|---------|---------|---------|---------|---------|---------|---------|
|            |                   | 1/2                                   | 1/3     | 1/4     | 1/5     | 1/6     | 1/7     | 1/8     | 1/9     |
| Original   | -                 | 0.00028                               | 0.00096 | 0.00081 | 0.00063 | 0.02556 | 0.06913 | 0.10572 | 0.03696 |
| Convolved1 | Average filter    | 0.07854                               | 0.14876 | 0.29312 | 0.47609 | 0.43306 | 1.06601 | 1.02536 | 1.25002 |
| Convolved2 | Sharpening filter | 1.34010                               | 4.49836 | 1.30898 | 3.54089 | 0.22995 | 2.66963 | 2.56190 | 3.47551 |
| Convolved3 | Laplacian filter  | 3.17143                               | 3.78570 | 4.49095 | 6.55959 | 3.33293 | 6.22306 | 5.28888 | 6.71930 |
| Convolved4 | Gaussian filter   | 0.07820                               | 0.14339 | 0.32122 | 0.53333 | 0.49635 | 1.13168 | 1.15413 | 2.09468 |

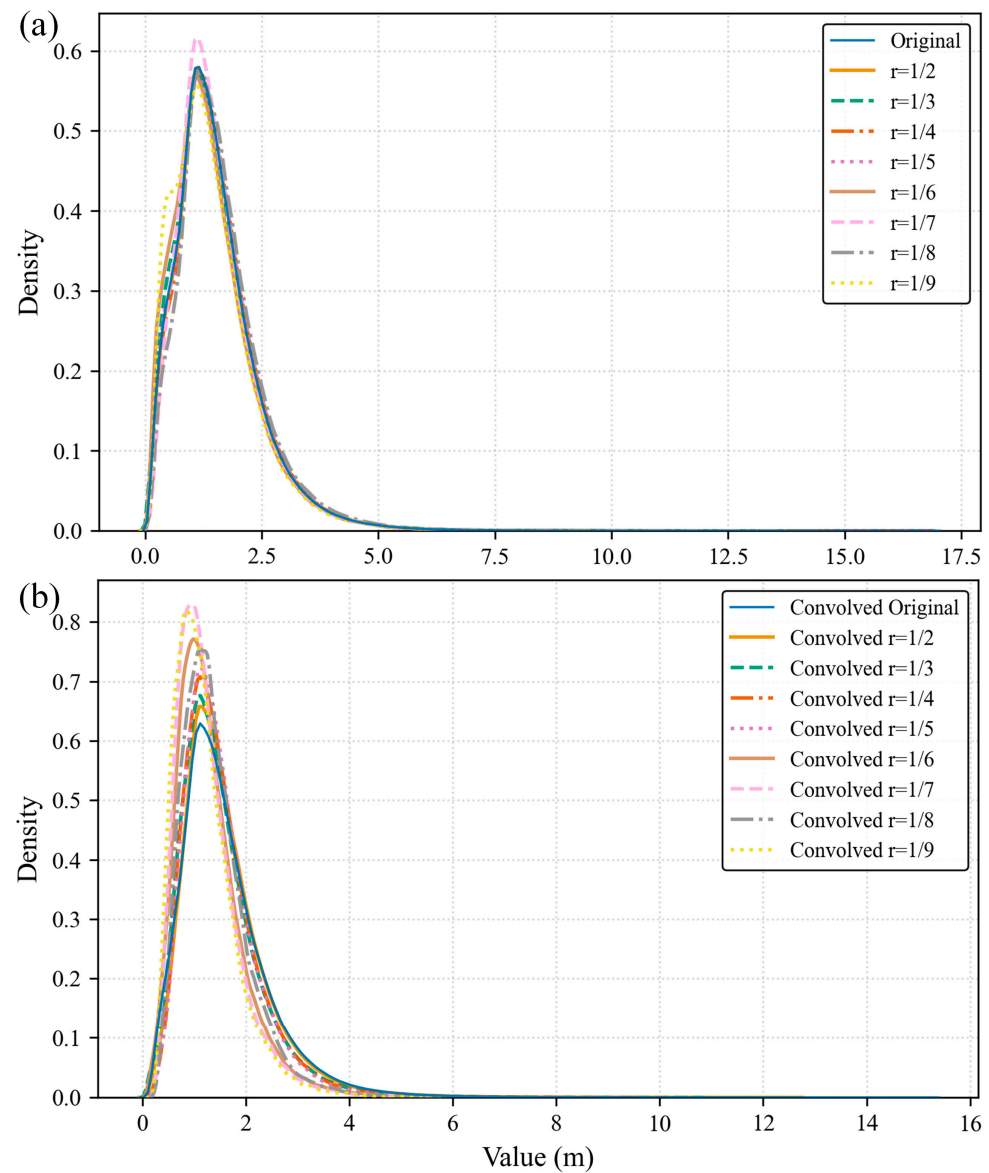


**Figure 6.** Comparison of KDE distributions. (a) Original data. (b) Data after  $3 \times 3$  average filter convolution.

### 3.3.1. Information Theory

Using information theory, we demonstrated that the KL divergences of the original data and those after the CS algorithm largely reflected the information content of the data distribution, with only minor differences. Assuming that the probability distribution of the original data is

$$p(x) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \delta(x - x_i) \quad (18)$$

where  $x_i$  is the pixel value, and sampling is uniform in the spatial dimension (let the sampling rate be  $r$ ,  $r = \frac{1}{2}$  represents sampling at every interval), then the local spatial distribution of the long statistical series should be largely steady [64], and the data should have the same local statistical characteristics. It follows that:

$$q_r(x) = \lim_{N \rightarrow \infty} \frac{1}{N_r} \sum_{j=1}^{N_r} \delta(x - x_{rj}) \quad \text{and} \quad (19)$$

$$\mathbb{P}(x_{rj} \in A) \approx \mathbb{P}(x_i \in A), \quad (20)$$

where  $N_r = \frac{N}{r}$ ,  $x_{rj}$  is the pixel value selected at sampling rate  $r$  and  $\mathbb{P}$  is the probability. Therefore, for an arbitrary set of measures  $A$ ,  $q_r(x) \approx p(x)$ ,  $r$  accounts for

$$\int p(x)dx \approx \int q_r(x)dx \quad (21)$$

When  $N$  is large enough, the sample histograms tend toward identical distributions [69]. According to the definition of KL divergence:

$$D_{\text{KL}}(p \parallel q_r) = \int p(x) \log \frac{p(x)}{q_r(x)} dx \quad (22)$$

Thus,  $\log \frac{p(x)}{q_r(x)} \approx 0$ , and the integration gives  $D_{\text{KL}}(p \parallel q_r) \approx 0$ . This explains why, regardless of the sampling rate from 1/2 to 1/9, the distribution  $q_r(x)$  will be very close to the original distribution  $p(x)$  as long as the data is steady and sufficiently smooth, generating KL divergences that are all close to zero. From an information-theory perspective, the entropy of the original data is

$$H(p) = - \int p(x) \log p(x) dx. \quad (23)$$

Whereas in the uniform sampling condition, the individual sampling distribution  $q_r(x)$  retains most of the information, the sampled mutual information is

$$I(p; q_r) = H(p) - D_{\text{KL}}(p \parallel q_r) \approx H(p). \quad (24)$$

**Interim Summary (3.3.1):** Assuming steady and uniform sampling, the sampling distribution  $q_r$  is highly similar to the original distribution  $p$  (see Equations (18)–(24)), resulting in  $D_{\text{KL}}(p \parallel q_r) \approx 0$ , and retains most of the mutual information. This indicates that compressed sampling itself does not fundamentally weaken the signal for learning in terms of information content.

### 3.3.2. Differential Amplification Effect

We demonstrated how convolution processing amplifies the subtle but informative differences: local enhancement kernels (or general learning-based convolution Jacobians) can amplify perturbations, and the amplifying effects of hierarchical stacking combined with gating/memory structures accumulate and amplify key features, enhancing the discriminatory power of weak information for downstream predictions.

Using Convolved2 as an example, the sharpening filter is expressed in the form [70]:

$$K_{\text{sharp}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (25)$$

Let the local region of the image be  $y(i, j)$ , then the output after convolution operation is:

$$y(i, j) = \sum_{m=-1}^1 \sum_{n=-1}^1 K_{\text{sharp}}(m, n) x(i+m, j+n) \quad (26)$$

Expanding this equation, we obtain

$$y(i, j) = 5x(i, j) - [x(i-1, j) + x(i+1, j) + x(i, j-1) + x(i, j+1)],$$

When this localized region is smooth:

$$x(i, j) \approx \frac{1}{4}[x(i-1, j) + x(i+1, j) + x(i, j-1) + x(i, j+1)], \quad (27)$$

at this point  $y(i, j) \approx 5x(i, j) - 4x(i, j) = x(i, j)$ .

When there are local edges or small differences, i.e.,  $x(i, j)$  differs from its neighbors,

$$\begin{aligned} \Delta y(i, j) &= y(i, j) - x(i, j) \\ &= 4[x(i, j) - \frac{1}{4}(x(i-1, j) + x(i+1, j) + x(i, j-1) + x(i, j+1))]. \end{aligned} \quad (28)$$

The right-hand form is proportional to the local gradient or difference (a discrete form of the Laplace operator that represents the difference between the prepixel value  $x(i, j)$  and its four neighboring pixel means). Thus, the sharpening kernel amplifies the local deviation from the mean by a factor of 4, allowing otherwise small differences to be more clearly expressed in the output.

In machine learning, convolutional kernels usually start with a random initialization and update the parameters through training. With generalization, the output is obtained when a local region  $x$  on the space is passed through the convolutional layer  $f_\theta$ :

$$y = f_\theta(x) \quad (29)$$

where  $\theta$  is a parameter of the convolution kernel and  $y$  is the desired output. This corresponds to a small perturbation  $\delta x$ , determined according to the Taylor expansion:

$$f_\theta(x + \delta x) \approx f_\theta(x) + J_{f_\theta}(x)\delta x, \quad (30)$$

where  $J_{f_\theta}(x) = \frac{\partial f_\theta}{\partial x}$  is the Jacobian matrix of the convolutional layer at  $x$  [65]. Using the loss function  $L(f_\theta(x), y)$ ,

$$\frac{\partial L(f_\theta(x), y)}{\partial x} = J_{f_\theta}(x)^T \frac{\partial L}{\partial f_\theta(x)} \quad (31)$$

If the training objective requires sensitivity to small changes,  $\delta x$  leads to large loss variations in training, and the corresponding gradient descent process adjusts the parameter  $\theta$  to capture subtle features. A Jacobian matrix with larger eigenvalues in the direction of this target is represented as:

$$\| \delta y \| \approx \| J_{f_\theta}(x)\delta x \| \gg \| \delta x \| \quad (32)$$

Thus,  $J_{f_\theta}(x)$  has a larger spectral paradigm in the target directions to amplify  $\delta x$ . If there are local subtle differences between the original and uniformly sampled data during training, and these subtle differences are highly relevant to the prediction task, gradient descent will make certain convolution kernel weights amplify the local differences, enhancing learning of the different features to improve the learning of key information.

**Interim Summary (3.3.2):** We demonstrated that convolution operators can differentially amplify small task-relevant local perturbations. The amplification factors were derived through the sharpening kernel examples (Equations (25)–(28)), and Taylor expansion and Jacobian matrix analysis (Equations (29)–(32)). We observed that with convolutional layers with large Jacobian eigenvalues in certain directions, small input perturbations  $\delta x$  were mapped to significantly amplified outputs  $\delta y$ . During training, gradient descent drives the convolutional kernel to enhance sensitivity in directions related to the loss, thereby amplifying the expression of originally weak features in the feature maps.

### 3.3.3. Amplification Effect Contributes to CSCL Accuracy

The CSCL comprises multiple layers of stacked cells, where the output of the  $l^{th}$  layer is

$$h^{(l)} = f^{(l)}(h^{(l-1)}). \quad (33)$$

Here, each layer  $f^{(l)}$  contains the convolution operation and gating mechanism. For a small perturbation  $\delta x$  of the initial input  $x = h^{(0)}$ , after the first layer

$$\delta h^{(1)} \approx J^{(1)}(x) \delta x. \quad (34)$$

Second layer:

$$\delta h^{(2)} \approx J^{(2)}(h^{(1)}) \delta h^{(1)} \approx J^{(2)}(h^{(1)}) J^{(1)}(x) \delta x. \quad (35)$$

To the  $L^{th}$  layer, it follows that:

$$\delta h^{(L)} \approx \left( \prod_{l=1}^L J^{(l)} \right) \delta x. \quad (36)$$

If at each layer, the network is tuned by training so that the local amplification factor  $\alpha_l$  in the direction of key features satisfies  $\alpha_l > 1$ , the total amplification effect is

$$\| \delta h^{(L)} \| \approx \left( \prod_{l=1}^L \alpha_l \right) \| \delta x \|. \quad (37)$$

This indicates that even if the initial small difference  $\delta x$  is small, it can be amplified exponentially after multi-layer stacking, improving the feature representation of the network as well as the model's sensitivity to local variations and generalizability.

Each unit of CSCL not only performs simple convolutional operations but also contains input gates, forgetting gates, output gates, and candidate memory units, such as Equation (6). When small local variations contribute to the final prediction, backpropagation drives  $i_t$  to approach 1 in these directions, and  $f_t$  retains the useful information to amplify the gradient transfer in these directions.

**Interim Summary (3.3.3):** Extending the conclusions from single-layer amplification to multi-layer stacks, the propagation of perturbations is determined based on the product of the Jacobian matrices across layers (Equations (33)–(37)). Therefore, if each layer has an amplification factor  $\alpha_l > 1$  in the target direction, the total amplification effect can accumulate multiplicatively ( $\prod \alpha_l$ ). Additionally, the gating and memory mechanisms in the CSCL unit (input, forget gate, output gate, and candidate memories) work together during forward propagation and backpropagation to retain and transmit amplified useful signals (e.g.,  $i_t \rightarrow 1$  in the beneficial direction,  $f_t$  retains information), further enhancing gradient flow and key feature representation.

**Final conclusion:** In conclusion, the information-preserving properties of uniform compression sampling ensure that the learner obtains key statistical information while the convolutional amplification-based training (in conjunction with gating/memory mechanisms) selectively enhances task-related perturbations. The combined effect enables CSCL to recover predictive clues from sparse samples, outperforming the original ConvLSTM in medium-range forecasting.

### 3.4. Model Parameter Setting

The CSCL model uses grid search to determine the optimal hyperparameters. The model was trained with a batch size of 256 and fixed number of 64 filters in the convolutional layers, each with a kernel size of  $3 \times 3$ , stride of  $1 \times 1$ , and “same” padding to preserve the

input spatial dimensions. Each convolutional layer was followed by batch normalization and ReLU activation, and max-pooling layers were applied using a  $2 \times 2$  kernel size and  $2 \times 2$  stride. The LSTM layer contained 256 hidden units, while the fully connected layer outputted 1681 neurons, corresponding to the spatial dimensions of the  $41 \times 41$  output. The training process aimed to minimize the MSE loss function, appropriate for regression tasks, and used Adam as the optimizer. The initial learning rate was set to  $1 \times 10^{-4}$ , with gradient clipping applied via a global L2 norm at a threshold of 1 to ensure training stability [71]. The learning rate was scheduled using ReduceLROnPlateau to improve convergence [72], reducing it by a factor of 0.5 when the validation loss plateaus for 10 consecutive epochs. This strategy ensures that the learning rate is adjusted in response to the progress of the model, avoiding premature convergence. Training was conducted for a maximum of 100 epochs, and the model stopped early if no improvement in validation loss occurred over successive epochs. For the weight initialization, the convolutional layers utilized Kaiming initialization (kaiming\_uniform\_) [73], and the gated recurrent unit layer weights were initialized using orthogonal initialization (orthogonal\_) [74]. Biases were initialized to 0. The entire model was trained using PyTorch 2.5.1 with CUDA 12.4, executed on an NVIDIA GeForce GTX 1080 Ti GPU (Santa Clara, CA, USA) to ensure efficient computation on large datasets.

## 4. Results and Discussion

### 4.1. SWH Prediction Results

Although the CSCL model effectively improves the prediction accuracy (as outlined in Section 3.3), further validation was required. Table 3 contains the results of several sets of ablation experiments conducted using the CSCL on SWH data.

**Table 3.** Error metrics of SWH prediction on test-set.

| Metrics | Time | ConvLSTM | CSCL-5,1 | CSCL-24,1 | CSCL-5,4 | CSCL-8,4 | CSCL-12,4 | CSCL-24,4 | CSCL-5,7 |
|---------|------|----------|----------|-----------|----------|----------|-----------|-----------|----------|
| RMSE    | 1 h  | 0.1095   | 0.0326   | 0.0325    | 0.0291   | 0.0263   | 0.0282    | 0.0316    | 0.0274   |
|         | 3 h  | 0.1726   | 0.0919   | 0.0844    | 0.0735   | 0.0683   | 0.0715    | 0.0762    | 0.0687   |
|         | 6 h  | 0.2531   | 0.1760   | 0.1701    | 0.1431   | 0.1352   | 0.1434    | 0.1515    | 0.1375   |
|         | 12 h | 0.3892   | 0.3418   | 0.3283    | 0.2615   | 0.2593   | 0.2806    | 0.2919    | 0.2569   |
|         | 18 h | 0.4878   | 0.4638   | 0.4528    | 0.3759   | 0.3756   | 0.3848    | 0.4105    | 0.3713   |
|         | 24 h | 0.5754   | 0.5482   | 0.5436    | 0.4722   | 0.4706   | 0.4840    | 0.5251    | 0.4632   |
| MAE     | 1 h  | 0.0694   | 0.0180   | 0.0182    | 0.0150   | 0.0133   | 0.0163    | 0.0167    | 0.0146   |
|         | 3 h  | 0.1136   | 0.0531   | 0.0503    | 0.0436   | 0.0402   | 0.0422    | 0.0445    | 0.0402   |
|         | 6 h  | 0.1706   | 0.1112   | 0.1025    | 0.0894   | 0.0856   | 0.0890    | 0.0937    | 0.0842   |
|         | 12 h | 0.2638   | 0.2211   | 0.2044    | 0.1684   | 0.1643   | 0.1795    | 0.1864    | 0.1655   |
|         | 18 h | 0.3294   | 0.3101   | 0.2915    | 0.2446   | 0.2418   | 0.2528    | 0.2617    | 0.2376   |
|         | 24 h | 0.3931   | 0.3730   | 0.3612    | 0.2999   | 0.3184   | 0.3295    | 0.3408    | 0.3008   |
| $R^2$   | 1 h  | 0.9763   | 0.9976   | 0.9975    | 0.9981   | 0.9985   | 0.9984    | 0.9979    | 0.9985   |
|         | 3 h  | 0.9431   | 0.9814   | 0.9845    | 0.9878   | 0.9897   | 0.9889    | 0.9879    | 0.9899   |
|         | 6 h  | 0.8810   | 0.9356   | 0.9412    | 0.9559   | 0.9613   | 0.9560    | 0.9528    | 0.9605   |
|         | 12 h | 0.7254   | 0.7776   | 0.7976    | 0.8651   | 0.8683   | 0.8478    | 0.8355    | 0.8688   |
|         | 18 h | 0.5782   | 0.6146   | 0.6302    | 0.7385   | 0.7378   | 0.7226    | 0.6928    | 0.7399   |
|         | 24 h | 0.4190   | 0.4757   | 0.4837    | 0.6025   | 0.6013   | 0.5596    | 0.5029    | 0.6030   |

We first analyzed the model input moment for ConvLSTM, CSCL-5,1, and CSCL-24,1, each with 1 layer and input of 1, 5, and 24 frames, respectively. Compared to the benchmark model ConvLSTM, the performance of CSCL-5,1 improved greatly due to the increase in the amount of input data, especially during 1–6 h, and less so above 12 h. The prediction accuracy improved slightly between CSCL-5,1 and CSCL-24,1.

We analyzed the input moment for CSCL with four layers. All models showed a significant improvement compared to the benchmark model ConvLSTM, both in terms of nowcasting and short-term forecasting abilities. Interestingly, CSCL-5,4 had the best performance (Table 4) despite having the least inputs and consuming the least time for training, which is surprising since more input data is generally more favorable for training. This implies that, for wave prediction, data within the first 5 h have stronger correlation and spatiotemporal continuity, and that irrelevant information in data older than 5 h interferes with training.

**Table 4.** Improvement of CSCL in SWH prediction on test-set.

| Metric                               | Time | CSCL-5,1 | CSCL-24,1 | CSCL-5,4 | CSCL-8,4 | CSCL-12,4 | CSCL-24,4 | CSCL-5,7 |
|--------------------------------------|------|----------|-----------|----------|----------|-----------|-----------|----------|
| Improvement in <i>RMSE</i>           | 1 h  | 70.2%    | 70.3%     | 73.4%    | 76.0%    | 74.2%     | 71.1%     | 75.0%    |
|                                      | 3 h  | 46.8%    | 51.1%     | 57.4%    | 60.4%    | 58.6%     | 55.9%     | 60.2%    |
|                                      | 6 h  | 30.5%    | 32.8%     | 43.5%    | 46.6%    | 43.3%     | 40.1%     | 45.7%    |
|                                      | 12 h | 12.2%    | 15.6%     | 32.8%    | 33.4%    | 27.9%     | 25.0%     | 34.0%    |
|                                      | 18 h | 4.9%     | 7.2%      | 22.9%    | 23.0%    | 21.1%     | 15.8%     | 23.9%    |
|                                      | 24 h | 4.7%     | 5.5%      | 17.9%    | 18.2%    | 15.9%     | 8.7%      | 19.5%    |
| Improvement in <i>MAE</i>            | 1 h  | 74.1%    | 73.8%     | 78.4%    | 80.8%    | 76.5%     | 75.9%     | 79.0%    |
|                                      | 3 h  | 53.3%    | 55.7%     | 61.6%    | 64.6%    | 62.9%     | 60.8%     | 64.6%    |
|                                      | 6 h  | 34.8%    | 39.9%     | 47.6%    | 49.8%    | 47.8%     | 45.1%     | 50.6%    |
|                                      | 12 h | 16.2%    | 22.5%     | 36.2%    | 37.7%    | 32.0%     | 29.3%     | 37.3%    |
|                                      | 18 h | 5.9%     | 11.5%     | 25.7%    | 26.6%    | 23.3%     | 20.6%     | 27.9%    |
|                                      | 24 h | 5.1%     | 8.1%      | 23.7%    | 19.0%    | 16.2%     | 13.3%     | 23.5%    |
| Improvement in <i>R</i> <sup>2</sup> | 1 h  | 2.1%     | 2.1%      | 2.2%     | 2.2%     | 2.2%      | 2.2%      | 2.2%     |
|                                      | 3 h  | 3.9%     | 4.2%      | 4.5%     | 4.7%     | 4.6%      | 4.5%      | 4.7%     |
|                                      | 6 h  | 5.8%     | 6.4%      | 7.8%     | 8.4%     | 7.8%      | 7.5%      | 8.3%     |
|                                      | 12 h | 6.7%     | 9.1%      | 16.1%    | 16.5%    | 14.4%     | 13.2%     | 16.5%    |
|                                      | 18 h | 5.9%     | 8.3%      | 21.7%    | 21.6%    | 20.0%     | 16.5%     | 21.9%    |
|                                      | 24 h | 11.9%    | 13.4%     | 30.5%    | 30.3%    | 25.1%     | 16.7%     | 30.5%    |

In terms of the different layers, both CSCL-5,4 and CSCL-5,7 generated more accurate forecasts than for ConvLSTM. Meanwhile, no significant difference was observed between CSCL-5,7 and CSCL-5,4, despite the variation in model layer complexity.

Figure 7 shows the scatter density plots of the predicted and true values.

The poorer results of ConvLSTM are reflected in the smaller predictions made for larger true values, especially during 12–24 h (Figure 7). In contrast, the optimal model CSCL-5,4 tended to generate predictions more similar to the true values, with a more coherent distribution of data. Finally, by fixing the number of model layers and increasing the input moments, as in the case of CSCL-5,4 at 24 h, the scatter plot distribution was almost similar to that of ConvLSTM. This suggests that, although the amount of data is substantially increased, the information is irrelevant for prediction and instead detrimental to model learning and prediction.

#### 4.2. MWP Prediction Results

We compared the performance of the benchmark model ConvLSTM and optimal model CSCL-5,4 for predicting MWP (Figure 8).

Compared with the SWH prediction, the scatter density plots showed more concentrated distributions with the MWP predictions. CSCL-5,4 still showed better performance than the baseline model ConvLSTM (Table 5).

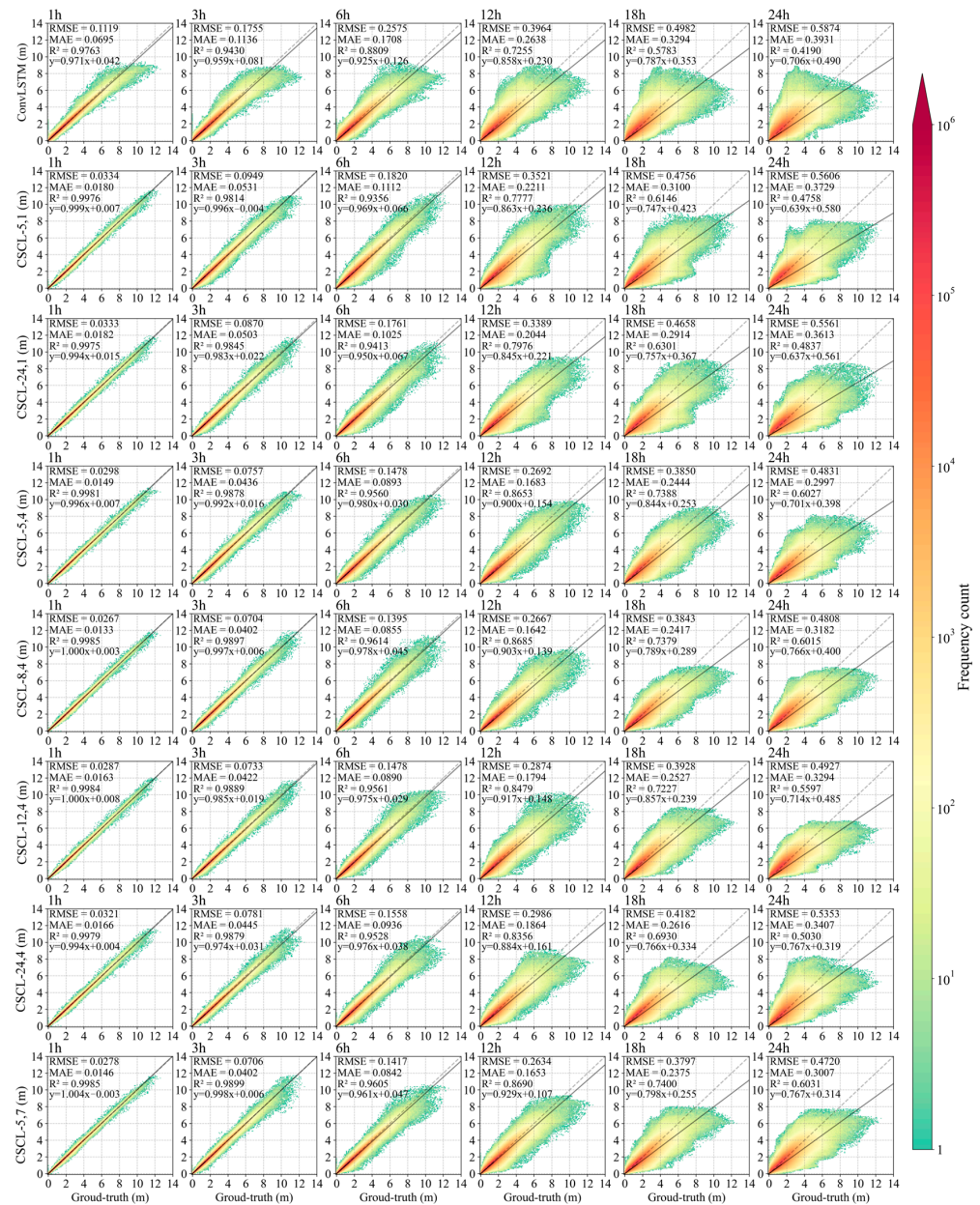


**Figure 7.** Scatter density plots for eight models in SWH prediction. (The solid line represents the 1:1 line, and the dashed line indicates the linear fit of the scatter points).

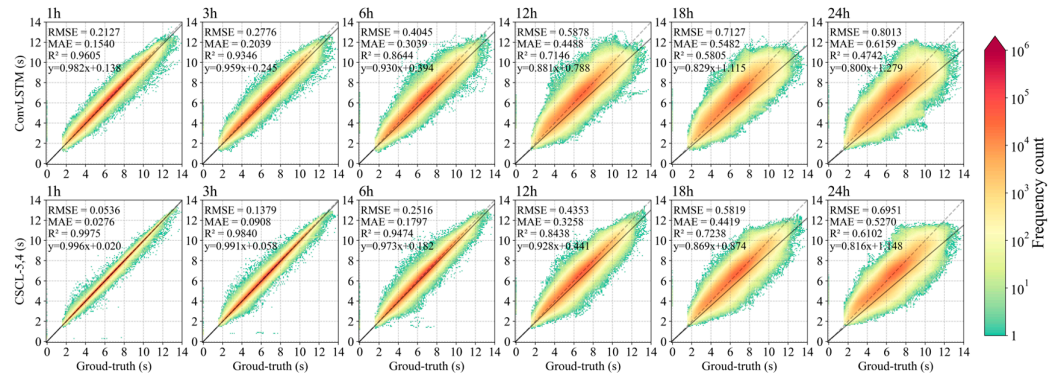


**Figure 8.** Scatter density plots in MWP prediction. (The solid line represents the 1:1 line, and the dashed line indicates the linear fit of the scatter points).

**Table 5.** Error metrics of MWP prediction on test-set.

| Metric         | Time | ConvLSTM | CSCL-5,4 | Improvement |
|----------------|------|----------|----------|-------------|
| RMSE           | 1 h  | 0.2127   | 0.0536   | 74.8%       |
|                | 3 h  | 0.2776   | 0.1379   | 50.3%       |
|                | 6 h  | 0.4045   | 0.2516   | 37.8%       |
|                | 12 h | 0.5878   | 0.4355   | 25.9%       |
|                | 18 h | 0.7127   | 0.5819   | 18.4%       |
|                | 24 h | 0.8013   | 0.6951   | 13.3%       |
| MAE            | 1 h  | 0.1540   | 0.0276   | 82.1%       |
|                | 3 h  | 0.2039   | 0.0908   | 55.5%       |
|                | 6 h  | 0.3039   | 0.1797   | 40.9%       |
|                | 12 h | 0.4488   | 0.3258   | 27.4%       |
|                | 18 h | 0.5482   | 0.4419   | 19.4%       |
|                | 24 h | 0.6159   | 0.5270   | 14.4%       |
| R <sup>2</sup> | 1 h  | 0.9605   | 0.9975   | 3.7%        |
|                | 3 h  | 0.9346   | 0.9840   | 5.0%        |
|                | 6 h  | 0.8644   | 0.9474   | 8.8%        |
|                | 12 h | 0.7146   | 0.8438   | 15.3%       |
|                | 18 h | 0.5805   | 0.7238   | 19.8%       |
|                | 24 h | 0.4742   | 0.6102   | 22.3%       |

#### 4.3. Discussion and Application

##### 4.3.1. Comparison with Advanced Models

We compared CSCL-5,4 with the advanced deterministic forecast models UNet and SmaAt-UNet in terms of SWH prediction (Table 6). The UNet model adopts the parameter settings of Lin, Tang, Wang, Wang, and Dong [75] (code available at <https://github.com/tensorflow/tensorflow>, accessed on 18 August 2025). The SmaAt-UNet model uses the parameter settings of Trebing, Stańczyk, and Mehrkanoon [76] (code available at <https://github.com/HansBambel/SmaAt-UNet>, accessed on 20 August 2025). CSCL-5,4 and SmaAt-UNet performed significantly better than ConvLSTM and UNet (Table 6). For the 12–24 h lead time, CSCL-5,4 performed slightly better than SmaAt-UNet.

**Table 6.** Error metrics for SWH prediction using CSCL-5,4 and other advanced methods.

| Metrics        | Time | ConvLSTM | CSCL-5,4 | UNet   | SmaAt-UNet |
|----------------|------|----------|----------|--------|------------|
| RMSE           | 1 h  | 0.1095   | 0.0291   | 0.0327 | 0.0308     |
|                | 3 h  | 0.1726   | 0.0735   | 0.0793 | 0.0787     |
|                | 6 h  | 0.2531   | 0.1431   | 0.1509 | 0.1468     |
|                | 12 h | 0.3892   | 0.2615   | 0.2827 | 0.2723     |
|                | 18 h | 0.4878   | 0.3759   | 0.4183 | 0.3915     |
|                | 24 h | 0.5754   | 0.4722   | 0.5109 | 0.4936     |
| MAE            | 1 h  | 0.0694   | 0.0150   | 0.0165 | 0.0154     |
|                | 3 h  | 0.1136   | 0.0436   | 0.0471 | 0.0468     |
|                | 6 h  | 0.1706   | 0.0894   | 0.1009 | 0.0936     |
|                | 12 h | 0.2638   | 0.1684   | 0.1846 | 0.1754     |
|                | 18 h | 0.3294   | 0.2446   | 0.2763 | 0.2607     |
|                | 24 h | 0.3931   | 0.2999   | 0.3280 | 0.3124     |
| R <sup>2</sup> | 1 h  | 0.9763   | 0.9981   | 0.9971 | 0.9974     |
|                | 3 h  | 0.9431   | 0.9878   | 0.9857 | 0.9864     |
|                | 6 h  | 0.8810   | 0.9559   | 0.9573 | 0.9642     |
|                | 12 h | 0.7254   | 0.8651   | 0.8462 | 0.8590     |
|                | 18 h | 0.5782   | 0.7385   | 0.7056 | 0.7243     |
|                | 24 h | 0.4190   | 0.6025   | 0.5583 | 0.5872     |

#### 4.3.2. Application in Wave Energy Prediction

According to Wan, Zhang, Meng, and Wang [77], the wave energy under deep water conditions can be calculated from SWH  $H_s$  and MWP  $T_e$ :

$$P = \frac{\rho g^2}{64\pi} H_s^2 T_e \approx 0.49 H_s^2 T_e, \quad (38)$$

where  $\rho$  is the density of seawater and  $g$  is the gravitational acceleration. Figure 9 presents the wave energy estimates.

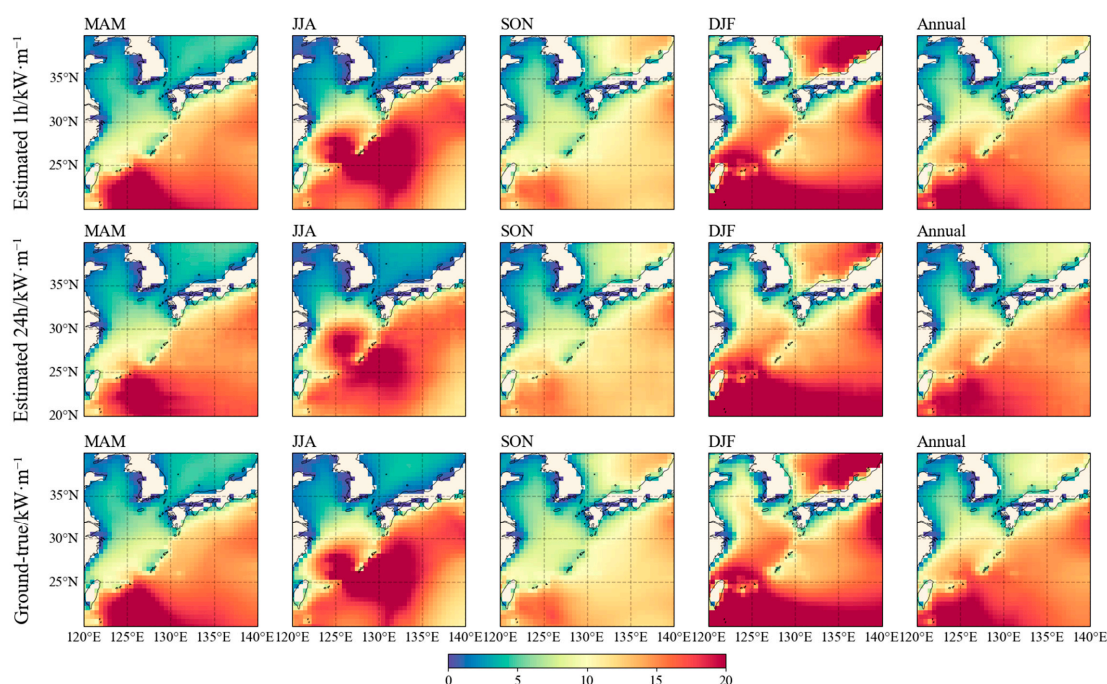


**Figure 9.** Comparison of CSCL results for wave energy estimation.

The CSCL model responded better to the distribution of wave energy for either annual or seasonal averages. For the whole year, wave energy was mainly concentrated in the Philippine Sea. Seasonally averaged spring (March–April–May) and winter (December–January–February) wave energy was also concentrated in the Philippine Sea, as well as during winter in the Sea of Japan. For summer (June–July–August), wave energy was mainly concentrated in the East China Sea and around the Ryukyu Islands. Finally, wave energy showed a reduced distribution in the fall (September–October–November).

These predictions can be used to select appropriate locations for wave energy generator installment between seasons to maximize energy generation. Otherwise, the eastern part of the Philippine Sea provides optimal conditions for fixed wave energy generation.

#### 4.3.3. Study Limitations

The scope of this study was limited to short-term forecasts with a lead time of 24 h, thus the generalizability of the CSCL for longer lead times (e.g., 48 h or more) requires further validation. Additionally, CSCL is a deterministic method and cannot output uncertainty measures (e.g., confidence intervals or probability fields), unlike probabilistic or ensemble forecasts, which may limit its direct interpretability in some contexts. Future research should evaluate the model under longer forecast lead times and extreme event scenarios, and explore avenues for probabilistic modeling or integration with ensemble methods.

## 5. Conclusions

In spatiotemporal series forecasting, current models show high accuracy for 0–12 h nowcasting but much lower accuracy for short-term 12–24 h forecasting. To address the limitations of current models, we focused on optimizing model structure and proposed the CSCL model for short-term forecasting.

Firstly, we considered the compressed sensing theory of signal processing in training with spatiotemporal series data. Because lossless compressive sensing is strongly dependent on data availability and certainty, which are limited in real-world applications, we adopted a simple uniform sampling method and incorporated data before and after sampling into model training. Since uniform sampling can preserve the data distribution by sampling in an infinite-length spatiotemporal series, it introduces a small loss in finite data. Therefore, CSCL not only utilizes the post-sampling data for training, but also gradually introduces the pre-sampling data, allowing the model to learn the effects of small differences between pre- and post-sampling and improve the prediction performance.

Secondly, we proved twice that the CSC operation amplifies the small sampling difference through theoretical inference and practical experimentation. This amplification greatly contributes to the model accuracy in the model training.

For SWH and MWP prediction, the model considering the first five frames of input data with four CSC layers (CSCL-5,4) had the best performance compared to the benchmark model ConvLSTM. Increasing the number of model layers and input data increased the training complexity, with minimal improvement in model accuracy.

Although we confirmed the performance of the CSCL model in a case study of wave energy prediction, further research is needed into the forecasting capability of the CSCL model for extreme events such as typhoons and tsunamis. The CSCL is expected to perform well in the forecasting of variables such as surface radiation, sea surface temperature, wind, pressure, and humidity. Most importantly, the study highlighted the importance of model input selection and structure optimization on prediction accuracy, providing a useful reference for further research on model inputs, such as U-V winds and sea level pressure.

**Author Contributions:** L.Z.: Conceptualization, Methodology, Writing—Original Draft Preparation, Investigation, Validation, Data curation. Y.K.: Formal Analysis, Investigation, Validation. J.Z.: Conceptualization, Supervision, Writing—Review and Editing, Funding Acquisition. B.T.: Supervision, Funding Acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Joint Program of the Science and Technology Plan of Liaoning Province (Natural Science Foundation—General Project, Project No. 2024-MSLH-057).

**Data Availability Statement:** The raw data for the study can be obtained from the ERA5 reanalysis produced by the European Centre for Medium-Range Weather Forecasts (available at: <https://cds.climate.copernicus.eu/datasets/reanalysis-era5-single-levels?tab=overview>, accessed on 15 February 2025). The dataset generated and analyzed in the current study is available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest. The sponsors had no role in the design, execution, interpretation, or writing of the study.

## References

1. Chen, P.; Wu, D.; Yang, Y.; Tsolakis, A. Discretization Modeling with Pseudo Real-Time Waves Input of the Hybrid Wave-Tidal Energy Converter based on Non-Linear Motions Rectification and Coupling Device. *Energy* **2025**, *323*, 135615. [CrossRef]
2. Alkhabbaz, A.; Hamzah, H.; Hamdoon, O.M.; Yang, H.-S.; Easa, H.; Lee, Y.-H. A unique design of a hybrid wave energy converter. *Renew. Energy* **2025**, *245*, 122814. [CrossRef]

3. Fanelli, E.; Masia, P.; Premici, A.; Volpato, E.; Da Ros, Z.; Aguzzi, J.; Francescangeli, M.; Dell'Anno, A.; Danovaro, R.; Cimino, R.; et al. The re-use of offshore platforms as ecological observatories. *Mar. Pollut. Bull.* **2024**, *209*, 117262. [\[CrossRef\]](#)
4. Cui, A.; Ma, Y.; Jiang, Y.; Li, S.; Zhang, J.; Wang, R. Synchronous inversion of bathymetry and wave height using wave textures and sun glint signals. *Ocean Eng.* **2025**, *328*, 121042. [\[CrossRef\]](#)
5. Li, C.; Xu, H.; Feng, X.; Sun, Q. Enhanced intelligent reconstruction study on wind wave height field in the South China Sea. *J. Ocean Eng. Sci.* **2025**. [\[CrossRef\]](#)
6. Zhao, L.; Li, Z.; Pei, Y.; Qu, L. Disentangled Seasonal-Trend representation of improved CEEMD-GRU joint model with entropy-driven reconstruction to forecast significant wave height. *Renew. Energy* **2024**, *226*, 120345. [\[CrossRef\]](#)
7. Cai, Z.; Zhang, B.-L.; Han, Q.-L.; Zhang, X.-M.; Zhang, J.; Xu, O. Sampled-data fuzzy modeling and control for offshore structures subject to parametric perturbations and wave loads. *Ocean Eng.* **2025**, *326*, 120908. [\[CrossRef\]](#)
8. Liang, J.; Fu, Y.; Wang, Y.; Ou, J. Identification of equivalent wind and wave loads for monopile-supported offshore wind turbines in operating condition. *Renew. Energy* **2024**, *237*, 121525. [\[CrossRef\]](#)
9. Song, Y.; Hong, X.; Zhang, Z.; Sun, T.; Cai, Y. Reliability analysis of floating offshore wind turbine considering multiple failure modes under extreme typhoon-wave condition. *Ocean Eng.* **2025**, *323*, 120564. [\[CrossRef\]](#)
10. Vandenhove, M.; Castelle, B.; Nicolae Lerma, A.; Marieu, V.; Martins, K.; Mazeiraud, V. Field measurements of wave and flow dynamics along a high-energy meso-macrotidal coast adjacent to a large estuary mouth. *Estuar. Coast. Shelf Sci.* **2025**, *317*, 109205. [\[CrossRef\]](#)
11. Fernández, L.; Calvino, C.; Dias, F. Sensitivity analysis of wind input parametrizations in the WAVEWATCH III spectral wave model using the ST6 source term package for Ireland. *Appl. Ocean Res.* **2021**, *115*, 102826. [\[CrossRef\]](#)
12. Jiang, Y.; Rong, Z.; Li, P.; Qin, T.; Yu, X.; Chi, Y.; Gao, Z. Modeling waves over the Changjiang River Estuary using a high-resolution unstructured SWAN model. *Ocean Model.* **2022**, *173*, 102007. [\[CrossRef\]](#)
13. Hoque, M.A.; Perrie, W.; Solomon, S.M. Application of SWAN model for storm generated wave simulation in the Canadian Beaufort Sea. *J. Ocean Eng. Sci.* **2020**, *5*, 19–34. [\[CrossRef\]](#)
14. Shankar, C.G.; Cambazoglu, M.K.; Bernstein, D.N.; Hesser, T.J.; Wiggert, J.D. Sensitivity and impact of atmospheric forcings on hurricane wind wave modeling in the Gulf of Mexico using nested WAVEWATCH III. *Appl. Ocean Res.* **2025**, *154*, 104320. [\[CrossRef\]](#)
15. Umesh, P.A.; Behera, M.R. Performance evaluation of input-dissipation parameterizations in WAVEWATCH III and comparison of wave hindcast with nested WAVEWATCH III-SWAN in the Indian Seas. *Ocean Eng.* **2020**, *202*, 106959. [\[CrossRef\]](#)
16. Zhao, L.; Li, Z.; Qu, L.; Zhang, J.; Teng, B. A hybrid VMD-LSTM/GRU model to predict non-stationary and irregular waves on the east coast of China. *Ocean Eng.* **2023**, *276*, 114136. [\[CrossRef\]](#)
17. Zhang, X.Y.; Li, Y.Q.; Gao, S.; Ren, P. Ocean Wave Height Series Prediction with Numerical Long Short-Term Memory. *J. Mar. Sci. Eng.* **2021**, *9*, 514. [\[CrossRef\]](#)
18. Yang, E. Analogy to numerical solution of wave propagation in an inhomogeneous medium with gain or loss variations. *Proc. IEEE* **1981**, *69*, 1574–1575. [\[CrossRef\]](#)
19. Holand, K.; Kalisch, H. Real-time ocean wave prediction in time domain with autoregression and echo state networks. *Front. Mar. Sci.* **2024**, *11*, 1486234. [\[CrossRef\]](#)
20. Zhao, L.; Li, Z.; Qu, L. Forecasting of Beijing PM2.5 with a hybrid ARIMA model based on integrated AIC and improved GS fixed-order methods and seasonal decomposition. *Heliyon* **2022**, *8*, e12239. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Alessio, S.; Longhetto, A.; Meixia, L. The Space and Time Features of Global SST Anomalies Studied by Complex Principal Component Analysis. *Adv. Atmos. Sci.* **1999**, *16*, 1–23. [\[CrossRef\]](#)
22. Majidian, H.; Enshaei, H.; Howe, D. A Concise Account for Challenges of Machine Learning in Seakeeping. *Procedia Comput. Sci.* **2025**, *253*, 2849–2858. [\[CrossRef\]](#)
23. Xia, Y.S.; Leung, H.; Chan, H. A prediction fusion method for reconstructing spatial temporal dynamics using support vector machines. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **2006**, *53*, 62–66. [\[CrossRef\]](#)
24. Chang, Z.H.; Liu, C.S.; Jia, J.M. STA-GCN: Spatial-Temporal Self-Attention Graph Convolutional Networks for Traffic-Flow Prediction. *Appl. Sci.* **2023**, *13*, 6796. [\[CrossRef\]](#)
25. Dai, K.; Li, X.T.; Ma, C.; Lu, S.Y.; Ye, Y.M.; Xian, D.; Tian, L.; Qin, D.Y. Learning Spatial-Temporal Consistency for Satellite Image Sequence Prediction. *IEEE Trans. Geosci. Remote. Sens.* **2023**, *61*, 3303947. [\[CrossRef\]](#)
26. Li, G.G.; Zhang, H.; Lyu, T.; Zhang, H.F. Regional significant wave height forecast in the East China Sea based on the Self-Attention ConvLSTM with SWAN model. *Ocean Eng.* **2024**, *312*, 119064. [\[CrossRef\]](#)
27. Deo, M.C.; Sridhar Naidu, C. Real time wave forecasting using neural networks. *Ocean Eng.* **1998**, *26*, 191–203. [\[CrossRef\]](#)
28. Makarynskyy, O.; Pires-Silva, A.A.; Makarynska, D.; Ventura-Soares, C. Artificial neural networks in wave predictions at the west coast of Portugal. *Comput. Geosci.* **2005**, *31*, 415–424. [\[CrossRef\]](#)
29. Yao, J.; Wu, W.H. Wave height forecast method with multi-step training set extension LSTM neural network. *Ocean Eng.* **2022**, *263*, 112432. [\[CrossRef\]](#)

30. Wang, M.; Ying, F.X. Point and interval prediction for significant wave height based on LSTM-GRU and KDE. *Ocean Eng.* **2023**, *289*, 116247. [\[CrossRef\]](#)
31. Neelamani, S. Influence of Threshold Value on Peak over Threshold Method on the Predicted Extreme Significant Wave Heights in Kuwaiti Territorial Waters. *J. Coast. Res.* **2009**, 564–568.
32. Stephens, S.A.; Gorman, R.M. Extreme wave predictions around New Zealand from hindcast data. *N. Z. J. Mar. Freshw. Res.* **2006**, *40*, 399–411. [\[CrossRef\]](#)
33. Zhao, L.; Li, Z.; Zhang, J.; Teng, B. An Integrated Complete Ensemble Empirical Mode Decomposition with Adaptive Noise to Optimize LSTM for Significant Wave Height Forecasting. *J. Mar. Sci. Eng.* **2023**, *11*, 435. [\[CrossRef\]](#)
34. Hisaki, Y. Wave hindcast in the North Pacific area considering the propagation of surface disturbances. *Prog. Oceanogr.* **2018**, *165*, 332–347. [\[CrossRef\]](#)
35. Wei, C.C.; Cheng, J.Y. Nearshore two-step typhoon wind-wave prediction using deep recurrent neural networks. *J. Hydroinformatics* **2020**, *22*, 346–367. [\[CrossRef\]](#)
36. Liu, W.C.; Huang, W.C. Hindcasting and predicting surge heights and waves on the Taiwan coast using a hybrid typhoon wind and tide-surge-wave coupled model. *Ocean Eng.* **2023**, *276*, 114208. [\[CrossRef\]](#)
37. Ti, Z.; Kong, Y. Single-instant spatial wave height forecast using machine learning: An image-to-image translation approach based on generative adversarial networks. *Appl. Ocean Res.* **2024**, *150*, 104094. [\[CrossRef\]](#)
38. Zhang, J.; Luo, F.; Quan, X.; Wang, Y.; Shi, J.; Shen, C.; Zhang, C. Improving wave height prediction accuracy with deep learning. *Ocean Model.* **2024**, *188*, 102312. [\[CrossRef\]](#)
39. Liu, Y.; Lu, W.; Wang, D.; Lai, Z.; Ying, C.; Li, X.; Han, Y.; Wang, Z.; Dong, C. Spatiotemporal wave forecast with transformer-based network: A case study for the northwestern Pacific Ocean. *Ocean Model.* **2024**, *188*, 102323. [\[CrossRef\]](#)
40. Wang, J.; Bethel, B.J.; Xie, W.; Dong, C. A hybrid model for significant wave height prediction based on an improved empirical wavelet transform decomposition and long-short term memory network. *Ocean Model.* **2024**, *189*, 102367. [\[CrossRef\]](#)
41. Jörges, C.; Berkenbrink, C.; Gottschalk, H.; Stumpe, B. Spatial ocean wave height prediction with CNN mixed-data deep neural networks using random field simulated bathymetry. *Ocean Eng.* **2023**, *271*, 113699. [\[CrossRef\]](#)
42. Yevnin, Y.; Chorev, S.; Dukan, I.; Toledo, Y. Short-term wave forecasts using gated recurrent unit model. *Ocean Eng.* **2023**, *268*, 113389. [\[CrossRef\]](#)
43. Meng, F.; Xu, D.; Song, T. ATDNNS: An adaptive time–frequency decomposition neural network-based system for tropical cyclone wave height real-time forecasting. *Future Gener. Comput. Syst.* **2022**, *133*, 297–306. [\[CrossRef\]](#)
44. Dixit, P.; Londhe, S.; Dandawate, Y. Removing prediction lag in wave height forecasting using Neuro—Wavelet modeling technique. *Ocean Eng.* **2015**, *93*, 74–83. [\[CrossRef\]](#)
45. Ti, Z.; Song, Y.; Deng, X. Spatial-temporal wave height forecast using deep learning and public reanalysis dataset. *Appl. Energy* **2022**, *326*, 120027. [\[CrossRef\]](#)
46. Ouyang, Z.; Zhao, Y.; Zhang, D.; Zhang, X. An effective deep learning model for spatial-temporal significant wave height prediction in the Atlantic hurricane area. *Ocean Eng.* **2025**, *317*, 120083. [\[CrossRef\]](#)
47. Luo, Y.; Shi, H.; Zhang, Z.; Zhang, C.; Zhou, W.; Pan, G.; Wang, W. Wave field predictions using a multi-layer perceptron and decision tree model based on physical principles: A case study at the Pearl River Estuary. *Ocean Eng.* **2023**, *277*, 114246. [\[CrossRef\]](#)
48. Price, I.; Sanchez-Gonzalez, A.; Alet, F.; Andersson, T.R.; El-Kadi, A.; Masters, D.; Ewalds, T.; Stott, J.; Mohamed, S.; Battaglia, P.; et al. Probabilistic weather forecasting with machine learning. *Nature* **2025**, *637*, 84–90. [\[CrossRef\]](#)
49. Krichen, M. Generative Adversarial Networks. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 6–8 July 2023; pp. 1–7.
50. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. In Proceedings of the 34th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 6–12 December 2020; Curran Associates Inc.: Vancouver, BC, Canada, 2020; pp. 6840–6851.
51. Wu, H.L.; Chung, W.Y. Sentiment-based masked language modeling for improving sentence-level valence-arousal prediction. *Appl. Intell.* **2022**, *52*, 16353–16369. [\[CrossRef\]](#)
52. Choi, B.; Jang, D.; Ko, Y. MEM-KGC: Masked Entity Model for Knowledge Graph Completion With Pre-Trained Language Model. *IEEE Access* **2021**, *9*, 132025–132032. [\[CrossRef\]](#)
53. Sun, X.; Zhang, J.; Croxford, A.J.; Drinkwater, B.W. A hardware compressed sensing method for ultrasonic imaging. *Sens. Actuators A Phys.* **2025**, *384*, 116265. [\[CrossRef\]](#)
54. Barnhill, D.; Yoshida, R.; Miura, K. Maximum inscribed and minimum enclosing tropical balls of tropical polytopes and applications to volume estimation and uniform sampling. *Comput. Geom.* **2025**, *128*, 102163. [\[CrossRef\]](#)
55. Niu, Y.; Li, H.; Tang, Z.; Liu, L.; Long, H.; Yan, H.; Zhu, M.; Zhang, J. STP-KDE: A spatiotemporal trajectory protection and publishing method based on kernel density estimation. *Comput. Electr. Eng.* **2024**, *117*, 109328. [\[CrossRef\]](#)

56. Rahad, M.; Shabab, R.; Ahammad, M.S.; Reza, M.M.; Karmaker, A.; Hossain, M.A. KL-FedDis: A federated learning approach with distribution information sharing using Kullback-Leibler divergence for non-IID data. *Neurosci. Inform.* **2025**, *5*, 100182. [\[CrossRef\]](#)
57. Waqas, M.; Humphries, U.W. A critical review of RNN and LSTM variants in hydrological time series predictions. *MethodsX* **2024**, *13*, 102946. [\[CrossRef\]](#) [\[PubMed\]](#)
58. Wang, J.; Li, X.; Li, J.; Sun, Q.; Wang, H. NGCU: A New RNN Model for Time-Series Data Prediction. *Big Data Res.* **2022**, *27*, 100296. [\[CrossRef\]](#)
59. Gao, B.; Xu, J.; Zhang, Z.; Liu, Y.; Chang, X. Marine diesel engine piston ring fault diagnosis based on LSTM and improved beluga whale optimization. *Alex. Eng. J.* **2024**, *109*, 213–228. [\[CrossRef\]](#)
60. Candes, E.J.; Wakin, M.B. An Introduction To Compressive Sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [\[CrossRef\]](#)
61. Rao, S. Satisfying the restricted isometry property with the optimal number of rows and slightly less randomness. *Inf. Process. Lett.* **2025**, *189*, 106553. [\[CrossRef\]](#)
62. Babu, S.; Aviyente, S.; Vaswani, N. Tensor Low Rank Column-Wise Compressive Sensing for Dynamic Imaging. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
63. Liu, X.; Wang, Z.; Ji, H.; Gong, H. Application and comparison of several adaptive sampling algorithms in reduced order modeling. *Heliyon* **2024**, *10*, e34928. [\[CrossRef\]](#)
64. Chen, J.; Song, P.; Zhao, C. Multi-scale self-supervised representation learning with temporal alignment for multi-rate time series modeling. *Pattern Recognit.* **2024**, *145*, 109943. [\[CrossRef\]](#)
65. Miki, T.; Chang, C.-W.; Ke, P.-J.; Telschow, A.; Tsai, C.-H.; Ushio, M.; Hsieh, C.-h. How to quantify interaction strengths? A critical rethinking of the interaction Jacobian and evaluation methods for non-parametric inference in time series analysis. *Phys. D Nonlinear Phenom.* **2025**, *476*, 134613. [\[CrossRef\]](#)
66. Zhao, L.; Li, Z.; Qu, L. A novel machine learning-based artificial intelligence method for predicting the air pollution index PM2.5. *J. Clean. Prod.* **2024**, *468*, 143042. [\[CrossRef\]](#)
67. Chisale, S.W.; Lee, H.S. Comprehensive onshore wind energy assessment in Malawi based on the WRF downscaling with ERA5 reanalysis data, optimal site selection, and energy production. *Energy Convers. Manag. X* **2024**, *22*, 100608. [\[CrossRef\]](#)
68. Dalla Torre, D.; Di Marco, N.; Menapace, A.; Avesani, D.; Righetti, M.; Majone, B. Suitability of ERA5-Land reanalysis dataset for hydrological modelling in the Alpine region. *J. Hydrol. Reg. Stud.* **2024**, *52*, 101718. [\[CrossRef\]](#)
69. Liaw, L.C.M.; Tan, S.C.; Goh, P.Y.; Lim, C.P. A histogram SMOTE-based sampling algorithm with incremental learning for imbalanced data classification. *Inf. Sci.* **2025**, *686*, 121193. [\[CrossRef\]](#)
70. Ai, X.; Ni, G.; Zeng, T. A sharpening median filter for Cauchy noise with wavelet based regularization. *J. Comput. Appl. Math.* **2025**, *467*, 116625. [\[CrossRef\]](#)
71. Łoś, M.; Służalec, T.; Paszyński, M.; Valseth, E. Stabilization of isogeometric finite element method with optimal test functions computed from L2 norm residual minimization. *J. Comput. Appl. Math.* **2025**, *460*, 116410. [\[CrossRef\]](#)
72. Khan, Z.; Liu, H.; Shen, Y.; Zeng, X. Deep learning improved YOLOv8 algorithm: Real-time precise instance segmentation of crown region orchard canopies in natural environment. *Comput. Electron. Agric.* **2024**, *224*, 109168. [\[CrossRef\]](#)
73. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1026–1034.
74. Chauhan, D.; Yadav, A. An archive-based self-adaptive artificial electric field algorithm with orthogonal initialization for real-parameter optimization problems. *Appl. Soft Comput.* **2024**, *150*, 111109. [\[CrossRef\]](#)
75. Lin, H.; Tang, J.; Wang, S.; Wang, S.; Dong, G. Deep learning downscaled high-resolution daily near surface meteorological datasets over East Asia. *Sci. Data* **2023**, *10*, 890. [\[CrossRef\]](#) [\[PubMed\]](#)
76. Trebing, K.; Stańczyk, T.; Mehrkanoon, S. SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture. *Pattern Recognit. Lett.* **2021**, *145*, 178–186. [\[CrossRef\]](#)
77. Wan, Y.; Zhang, J.; Meng, J.; Wang, J. Exploitable wave energy assessment based on ERA-Interim reanalysis data—A case study in the East China Sea and the South China Sea. *Acta Oceanol. Sin.* **2015**, *34*, 143–155. [\[CrossRef\]](#)

**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.