energies

MDPI

Article

Consistent and Orthogonal Multi-Scale Time–Frequency Feature Learning for Wind Speed Forecasting

Yun Wang, Wentao Liu and Fan Zhang *

School of Automation, Central South University, Changsha 410083, China; wangyun19@csu.edu.cn (Y.W.); lwtcsu28@csu.edu.cn (W.L.)
* Correspondence: zhangfan219@csu.edu.cn

Abstract

Reliable wind speed forecasting is essential to power dispatch and grid stability. However, the inherent variability and uncertainty of wind speed make accurate forecasting difficult for conventional single-scale and single-domain models. To better capture complex wind dynamics, recent studies have increasingly adopted multi-scale and time–frequency forecasting frameworks. Nevertheless, these methods often suffer from feature redundancy across scales and noise interference during cross-domain feature fusion. This study proposes Consistent and Orthogonal Multi-Scale Time–Frequency Feature Learning (COMTFFL), a model for wind speed forecasting. COMTFFL employs hierarchical downsampling and time–frequency transformations to construct multi-scale representations in both time and frequency domains. A dual-branch encoder is then used to extract temporal and spectral representations, which are subsequently fused and fed into a forecasting decoder for multi-step forecasting. To further improve feature learning, a composite Consistency–Orthogonality Constrained Forecasting (COCF) loss is introduced. Specifically, the consistency loss encourages alignment between pooled time-domain and frequency-domain representations at the same scale, while the orthogonality loss guides the mean cosine similarity between adjacent scales within each domain toward a specified target. Experiments on three historical wind speed datasets evaluate forecasting performance and the contribution of the proposed loss.

Keywords: wind speed forecasting; multi-scale time–frequency features; intra-scale cross-domain consistency; cross-scale intra-domain orthogonality

1. Introduction

Wind energy has become an important component of modern power systems due to its environmental benefits and abundant availability [1]. According to the International Renewable Energy Agency, global installed wind power capacity has continued to increase in recent years [2]. However, the variability and uncertainty of wind speed make power generation difficult to forecast, posing challenges to grid operation and wind power dispatching [3,4]. Therefore, accurate wind speed forecasting is essential to the reliable and efficient integration of wind power into power systems.

1.1. Literature Review

To accurately describe the complex fluctuation of wind speed, many researchers have proposed various forecasting methods. According to the forecasting horizon, wind speed forecasting can generally be classified into four categories: long-term (24 h to more than

Check for updates

Academic Editor: Lars Johanning

Received: 19 October 2026
Revised: 13 November 2026
Accepted: 18 November 2026
Published: 22 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

one week ahead), medium-term (6 to 24 h ahead), short-term (30 min to 6 h ahead), and ultra-short-term (a few seconds to 30 min ahead) [5]. Long-term forecasts are mainly used for maintenance planning and energy management, while medium-term forecasts support day-ahead scheduling and electricity market operations. Short-term forecasts provide guidance for intra-day dispatch, whereas ultra-short-term forecasts are important for real-time turbine control and load management.

Based on underlying modeling theories, wind speed forecasting methods are broadly divided into four main classes: physical, statistical, artificial intelligence (AI), and hybrid methods [6]. Physical methods, such as numerical weather prediction, rely on complex atmospheric physics equations [3]. However, they are computationally intensive and are generally more suitable for medium- and long-term forecasting. Conversely, statistical methods leverage historical meteorological data under linear assumptions for short-term forecasting, encompassing classic models like autoregressive, autoregressive moving average, and autoregressive integrated moving average (ARIMA) [7]. To better address the non-stationary nature of wind speed, specialized statistical variants such as RWT-ARIMA [8] and nonlinear autoregressive models [7] have been subsequently developed, effectively utilizing advanced techniques to capture complex dynamics and improve forecasting performance.

AI methods have become increasingly popular in wind speed forecasting because of their strong nonlinear modeling capability [9]. Early studies mainly employed shallow learning models, such as artificial neural networks (ANNs) [10], support vector machines (SVMs) [11], and adaptive neuro-fuzzy inference systems (ANFISs) [12]. Subsequently, deep learning models, recurrent neural networks (RNNs), long short-term memory (LSTM) networks [13,14], convolutional neural networks (CNNs), and temporal convolutional networks (TCNs) have been widely applied to wind speed forecasting [15–17]. These models have demonstrated improved performance by learning hierarchical representations directly from historical data. More recently, Transformer-based models have attracted increasing attention in wind speed forecasting [18,19]. Benefiting from the self-attention mechanism, Transformers are able to model long-range dependencies without relying on recurrent structures. Representative models such as TimesNet [20], iTransformer [21] and FITS [22] have demonstrated strong performance in general time-series forecasting, while AFMT [23] and WindTrans [24] have further validated the effectiveness of Transformer architectures in wind speed forecasting. Nevertheless, issues such as feature redundancy, limited interpretability, and performance degradation under different operating conditions still present challenges for practical applications.

Hybrid methods have been widely used in wind speed forecasting because they combine the strengths of different methods [25]. A common strategy is to first decompose the original wind speed series into several components and then model each component separately. For example, wavelet transform, empirical mode decomposition, and variational mode decomposition have been widely employed to reduce data non-stationarity and improve forecasting accuracy [26]. After decomposition, machine learning or deep learning models are usually applied to predict each component, and the individual forecasts are subsequently aggregated to obtain the final result [27–30]. Besides decomposition-based approaches, feature fusion has also attracted increasing attention in recent years. Unlike decomposition methods, these approaches focus on integrating information from different data sources, feature spaces, or temporal scales. By jointly exploiting multiple types of information, feature fusion methods have shown promising performance in capturing complex temporal dependencies and improving forecasting robustness [31–33].

However, many existing studies rely primarily on single-domain or single-scale features, which may limit their ability to capture complementary information across different representation spaces. To address this issue, increasing attention has been devoted to dual-

domain and multi-scale feature learning. For example, ref. [31] developed a multi-feature and multi-scale ensemble method (MFMS) using a dual-branch architecture, and [32] proposed a deep time–frequency augmented (DTFA) model. Ref. [34] introduced TFMSNet to concurrently extract multi-granularity temporal features and frequency components, while [35] designed a hierarchical dual-tower Transformer (Hidformer) to parallelize the time and frequency domains, and ref. [36] achieved multi-scale temporal–frequency fusion utilizing MLP-based architectures. Additionally, ref. [37] proposed MGCNet to align multi-scale forecasting targets via cross-attention, while [38] integrated MVMD with a restricted Crossformer for multi-frequency forecasting. These studies demonstrate the potential of integrating multi-scale analysis with time–frequency representations for forecasting tasks.

In summary, recent studies have increasingly combined advanced forecasting models with feature processing techniques to improve wind speed forecasting performance. By incorporating time–frequency analysis and multi-scale decomposition, these methods are able to capture both long-term trends and short-term fluctuations in wind speed series, leading to more informative feature representations.

1.2. Motivation and Contribution

Through the above analysis, two limitations of existing methods can be identified. First, the integration of time-domain and frequency-domain features may introduce inconsistencies between different feature representations, which can affect subsequent information fusion. Second, multi-scale features often contain redundant information, and the relationships among different scales are not explicitly considered in many existing studies. As a result, the learned features may contain both noise and redundant information, which can adversely affect forecasting performance.

To address the above limitations, this study proposes Consistent and Orthogonal Multi-Scale Time–Frequency Feature Learning (COMTFFL), a model for wind speed forecasting. COMTFFL applies multi-scale processing and time–frequency transformation to generate multi-scale representations in the time and frequency domains. A dual-branch encoder is then employed to extract temporal and spectral features, which are subsequently fused and fed into a sequence decoder for forecasting. To further improve feature learning, a Consistency–Orthogonality Constrained Forecasting (COCF) loss is introduced. Specifically, the consistency loss penalizes discrepancies between pooled time-domain and frequency-domain feature vectors at the same scale, whereas the orthogonality loss regulates cosine similarities between adjacent scales within each domain. The main contributions of this study are summarized as follows:

- A multi-scale time–frequency feature learning model, named COMTFFL, is developed to fuse multi-scale representations from both time and frequency domains for wind speed forecasting.
- An intra-scale cross-domain consistency loss penalizes the mean absolute difference between the pooled time-domain and frequency-domain representations at each scale.
- A cross-scale intra-domain orthogonality loss encourages the mean cosine similarity of adjacent scales to approach a specified target.
- Forecasting and ablation experiments on three wind speed datasets assess the architecture and its training constraints.

1.3. Organization of This Study

The remainder of this paper is organized as follows: Section 2 presents the architecture of the proposed model and the proposed composite loss function. Section 3 describes the implementation process of COMTFFL for wind speed forecasting. Section 4 presents and

analyzes the experimental results and discusses the effectiveness of the proposed modules and loss. Finally, Section 5 concludes the paper.

2. Proposed Wind Speed Forecasting Model—COMTFFL

This section presents the proposed model in detail; it consists of three primary modules: a Multi-scale Data Processing Module, a Multi-Scale Time–Frequency Feature Extraction Module, and a Multi-Scale Feature Fusion and Forecasting Module. The overall architecture of the proposed model is illustrated in Figure 1.

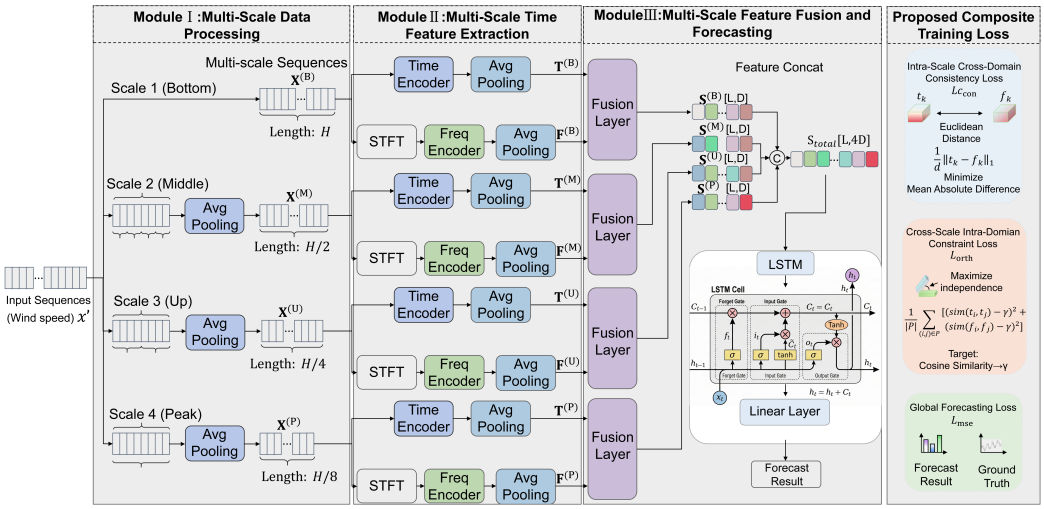

Figure 1. Structure of the proposed model, COMTFFL. The time-domain and frequency-domain encoders are shown in blue and green, respectively, while the fusion layers are shown in purple.

2.1. Module I: Multi-Scale Data Processing

A single sample for model training consists of a continuous wind speed sequence with length $H + L$, where H denotes the length of the input historical segment and L denotes the length of the target future segment. The complete sample sequence $\boldsymbol{x}$ can be expressed as:

$$\boldsymbol{x} = \{x_{t-H+1}, \ldots, x_{t-1}, x_t, \ldots, x_{t+L}\} \in \mathbb{R}^{H+L}, \tag{1}$$

where x_t represents the observed wind speed at time t. This study takes the H-length segment $\boldsymbol{x}^{'} = [x_{t-H+1}, \ldots, x_{t-1}, x_t] \in \mathbb{R}^H$ before time t as model input to perform L-step-ahead wind speed forecasting.

Multi-scale analysis uncovers distinct fluctuation patterns inherent in the data, thereby overcoming the information incompleteness of single-scale representations. To extract a hierarchical multi-scale representation from $\boldsymbol{x}^{'}$, this study utilizes a set of scale factors $S = \{s_0, s_1, s_2, s_3\} = \{1, 2, 4, 8\}$. This approach generates sequences at four distinct scales—denoted the Bottom (B), Middle (M), Up (U), and Peak (P) layers—via non-overlapping mean pooling with scale factors $s \in S$. This hierarchical structure effectively captures temporal patterns ranging from local fine-grained details to global semantic contexts. Formally, the downsampling function $F_{ds} : \mathbb{R}^H \to \mathbb{R}^{H/s}$ is defined as:

$$\mathbf{X}^{(s)} = F_{ds}\left(\boldsymbol{x}^{'}, s\right) \in \mathbb{R}^{1 \times \frac{H}{s}}, \tag{2}$$

$$x_i^{(s)} = \frac{1}{s} \sum_{j=0}^{s-1} x_{t-H+1+(i-1)s+j} \tag{3}$$

where $x_i^{(s)}$ is the i-th element of the downsampled sequence $\mathbf{X}^{(s)}$. The averaging window and stride are both equal to s. In this study, the input length $H = 48$ is divisible by all four scale factors $s \in \{1, 2, 4, 8\}$, giving sequences of 48, 24, 12, and 6 samples, respectively. Each average is computed over a complete group of input samples. For an input length that is not divisible by s, the implementation discards the incomplete group at the end of the sequence. The four scale sequences are given by:

$$\mathbf{X}^{(\mathrm{B})} = F_{ds}\left(\boldsymbol{x}', s_0\right) \in \mathbb{R}^{1\times H}, \tag{4}$$

$$\mathbf{X}^{(\mathrm{M})} = F_{ds}\left(\boldsymbol{x}', s_1\right) \in \mathbb{R}^{1\times \frac{H}{2}}, \tag{5}$$

$$\mathbf{X}^{(\mathrm{U})} = F_{ds}\left(\boldsymbol{x}', s_2\right) \in \mathbb{R}^{1\times \frac{H}{4}}, \tag{6}$$

$$\mathbf{X}^{(\mathrm{P})} = F_{ds}\left(\boldsymbol{x}', s_3\right) \in \mathbb{R}^{1\times \frac{H}{8}}. \tag{7}$$

The final output of the Multi-scale Data Processing Module is the set of these multi-scale sequences, denoted by $\mathbf{X}_{\mathrm{MS}}$:

$$\mathbf{X}_{\mathrm{MS}} = \left\{\mathbf{X}^{(\mathrm{B})}, \mathbf{X}^{(\mathrm{M})}, \mathbf{X}^{(\mathrm{U})}, \mathbf{X}^{(\mathrm{P})}\right\}. \tag{8}$$

These multi-scale sequences $\mathbf{X}_{\mathrm{MS}}$ serve as the foundation for the subsequent feature representation modules. Module I is illustrated in Figure 2.

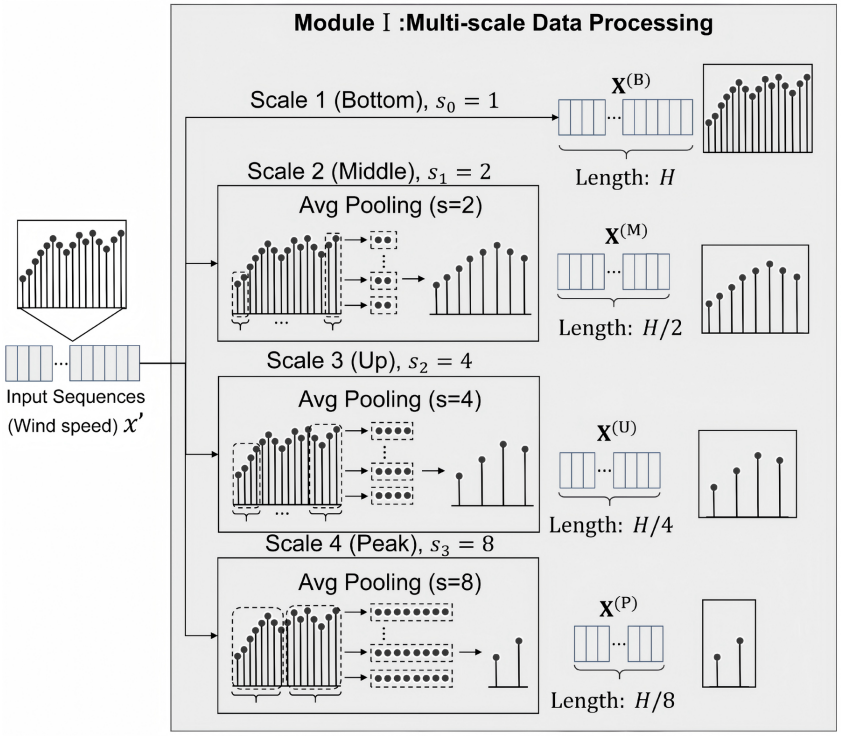

Figure 2. Structure of Module I.

2.2. Module II: Multi-Scale Time–Frequency Feature Extraction

As shown in Figure 3, Module II is composed of a Transformer-based Time-Domain Feature Extraction Module and an STFT-enhanced Frequency-Domain Feature Extraction Module. The dual-branch architecture is developed to capture multi-scale representations across both domains and align dimensionally for subsequent fusion.

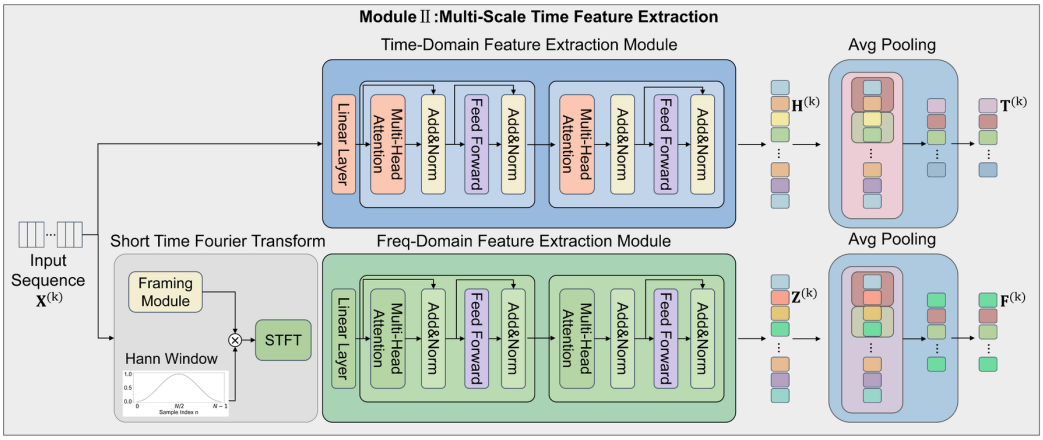

Figure 3. Structure of Module II. The time-domain and frequency-domain feature extraction modules are shown in blue and green, respectively.

2.2.1. Multi-Scale Time-Domain Feature Extraction

The Time-Domain Feature Extraction Module processes the hierarchical sequences $\mathbf{X}_{\text{MS}} = \left\{\mathbf{X}^{(\text{B})}, \mathbf{X}^{(\text{M})}, \mathbf{X}^{(\text{U})}, \mathbf{X}^{(\text{P})}\right\}$ derived in Section 2.1 in parallel. Taking the Middle-scale sequence $\mathbf{X}^{(\text{M})} \in \mathbb{R}^{1\times H/2}$ as a representative example, the time-domain feature representation process is formulated as follows. The tensor is first processed by a time-domain encoder $F^{(\text{M})}_{\text{time}}(\cdot)$, which consists of a Transformer-based architecture designed to capture complex temporal dependencies. Formally:

$$\mathbf{H}^{(\text{M})} = F^{(\text{M})}_{\text{time}}\left(\mathbf{X}^{(\text{M})}\right) \in \mathbb{R}^{\frac{H}{2}\times d_{\text{time}}}, \tag{9}$$

where d_{time} denotes the hidden feature dimension.

To unify the temporal dimensions across different scales for subsequent fusion, adaptive average pooling $\mathcal{P}_{\text{avg}}(\cdot)$ maps the encoded sequence to $L_{\text{seq}} = \lfloor H/4 \rfloor = 12$ for $H = 48$:

$$\mathbf{T}^{(\text{M})} = \mathcal{P}_{\text{avg}}\left(\mathbf{H}^{(\text{M})}\right) \in \mathbb{R}^{L_{\text{seq}}\times d_{\text{time}}}. \tag{10}$$

The same procedure is applied to all four scales, yielding the final time-domain feature representations:

$$\mathbf{T}^{(\text{B})} = \mathcal{P}_{\text{avg}}\left(F^{(\text{B})}_{\text{time}}(\mathbf{X}^{(\text{B})})\right) \in \mathbb{R}^{L_{\text{seq}}\times d_{\text{time}}}, \tag{11}$$

$$\mathbf{T}^{(\text{M})} = \mathcal{P}_{\text{avg}}\left(F^{(\text{M})}_{\text{time}}(\mathbf{X}^{(\text{M})})\right) \in \mathbb{R}^{L_{\text{seq}}\times d_{\text{time}}}, \tag{12}$$

$$\mathbf{T}^{(\text{U})} = \mathcal{P}_{\text{avg}}\left(F^{(\text{U})}_{\text{time}}(\mathbf{X}^{(\text{U})})\right) \in \mathbb{R}^{L_{\text{seq}}\times d_{\text{time}}}, \tag{13}$$

$$\mathbf{T}^{(\text{P})} = \mathcal{P}_{\text{avg}}\left(F^{(\text{P})}_{\text{time}}(\mathbf{X}^{(\text{P})})\right) \in \mathbb{R}^{L_{\text{seq}}\times d_{\text{time}}}, \tag{14}$$

The representations are expressed with the sequence dimension first, giving L_{seq} positions with $d_{\text{time}} = 64$ features per position.

2.2.2. Multi-Scale Frequency-Domain Feature Extraction

The short-time Fourier transform (STFT) is applied to the four sequences produced by Module I at the Bottom, Middle, Upper, and Peak scales, denoted by $\mathbf{X}^{(\text{B})}$, $\mathbf{X}^{(\text{M})}$, $\mathbf{X}^{(\text{U})}$, and $\mathbf{X}^{(\text{P})}$, respectively. Taking the Middle-scale sequence $\mathbf{X}^{(\text{M})} \in \mathbb{R}^{1\times H/2}$ as a representative example, let $x^{(\text{M})}[n]$ denote the discrete signal corresponding to $\mathbf{X}^{(\text{M})}$. Before applying the STFT, the input at each scale is padded on the right with zeros to a length of 64. The centered STFT then applies the reflection padding of 32 samples on each side. For the

Middle scale, let $\widetilde{x}^{(\mathrm{M})}$ denote the resulting 128-sample sequence, indexed from zero. The complex spectrum at frame m and frequency bin f is computed as

$$\mathrm{STFT}\{x^{(\mathrm{M})}\}[m,f] = \sum_{n=0}^{63} \widetilde{x}^{(\mathrm{M})}[mh+n]\, w[n]\, \exp\!\left(-\mathrm{j}\frac{2\pi f n}{64}\right), \tag{15}$$

where $w[n]$ is a 64-sample periodic Hann window and $h = 16$ is the hop length in samples at each scale. The same padding and framing procedure is applied at all four scales, giving $M_k = 1 + \lfloor (128-64)/16 \rfloor = 5$ frames per scale. The one-sided magnitude spectrogram contains $F_{\mathrm{bins}} = 33$ frequency bins per frame. For the Middle scale, $\mathbf{S}^{(\mathrm{M})} \in \mathbb{R}^{5\times 33}$.

For scale factors 1, 2, 4, and 8, the numbers of padded positions in the five STFT frames are $(0,0,16,31,31)$, $(17,24,40,56,64)$, $(41,41,52,64,64)$, and $(53,53,58,64,64)$, respectively. Each frame contains 64 positions, and the counts include padding zeros copied by reflection. The last frame at scale factor 2 and the last two frames at scale factors 4 and 8 contain only padding zeros.

A linear layer projects the 33 frequency bins in each frame to d_{freq} features before the frequency-domain Transformer:

$$\mathbf{Z}^{(\mathrm{M})}_{\mathrm{proj}} = \mathbf{S}^{(\mathrm{M})}\mathbf{W}_{\mathrm{freq}} + \mathbf{b}_{\mathrm{freq}}, \qquad \mathbf{W}_{\mathrm{freq}} \in \mathbb{R}^{F_{\mathrm{bins}}\times d_{\mathrm{freq}}}, \tag{16}$$

where $d_{\mathrm{freq}} = 64$ is the hidden dimension. The Transformer encoder processes $\mathbf{Z}^{(\mathrm{M})}_{\mathrm{proj}}$ along the STFT frame axis within each sample and returns $\mathbf{Z}^{(\mathrm{M})} \in \mathbb{R}^{M_k\times d_{\mathrm{freq}}}$.

To unify the temporal length across scales for subsequent fusion, adaptive average pooling is applied along the time-frame dimension in two stages. The frequency encoder first maps the $M_k = 5$ STFT frame features to $L_{\mathrm{enc}} = 32$ positions. The main model then maps them to the common fusion length $L_{\mathrm{seq}} = \lfloor H/4 \rfloor = 12$ for $H = 48$. Let $\mathcal{P}_{\mathrm{avg}}$ denote these two pooling stages in sequence. For the Middle scale, the resulting representation is

$$\mathbf{F}^{(\mathrm{M})} = \mathcal{P}_{\mathrm{avg}}\left(\mathbf{Z}^{(\mathrm{M})}\right) \in \mathbb{R}^{L_{\mathrm{seq}}\times d_{\mathrm{freq}}}. \tag{17}$$

Let $\mathcal{F}_{\mathrm{freq}}(\cdot)$ denote the sequence of operations from the STFT to the final pooling stage. The resulting frequency-domain features at the four scales are given by

$$\mathbf{F}^{(\mathrm{B})} = \mathcal{F}_{\mathrm{freq}}\left(\mathbf{X}^{(\mathrm{B})}\right) \in \mathbb{R}^{L_{\mathrm{seq}}\times d_{\mathrm{freq}}}, \tag{18}$$

$$\mathbf{F}^{(\mathrm{M})} = \mathcal{F}_{\mathrm{freq}}\left(\mathbf{X}^{(\mathrm{M})}\right) \in \mathbb{R}^{L_{\mathrm{seq}}\times d_{\mathrm{freq}}}, \tag{19}$$

$$\mathbf{F}^{(\mathrm{U})} = \mathcal{F}_{\mathrm{freq}}\left(\mathbf{X}^{(\mathrm{U})}\right) \in \mathbb{R}^{L_{\mathrm{seq}}\times d_{\mathrm{freq}}}, \tag{20}$$

$$\mathbf{F}^{(\mathrm{P})} = \mathcal{F}_{\mathrm{freq}}\left(\mathbf{X}^{(\mathrm{P})}\right) \in \mathbb{R}^{L_{\mathrm{seq}}\times d_{\mathrm{freq}}}, \tag{21}$$

All four frequency-domain representations have the same sequence length $L_{\mathrm{seq}} = 12$. At each scale, the frequency-domain representation is fused with the corresponding time-domain representation.

2.3. Module III: Multi-Scale Feature Fusion and Forecasting

Module III consists of an intra-scale time–frequency feature fusion layer, a cross-scale feature concatenation, and an LSTM-based prediction head, as shown in Figure 4.

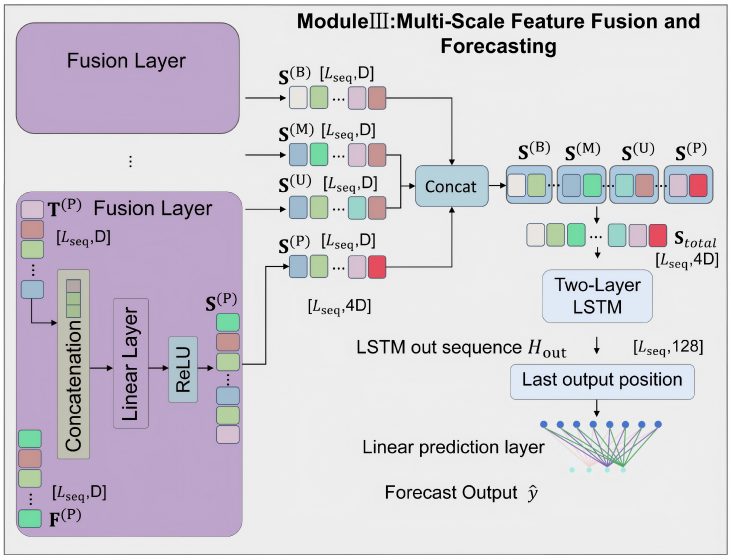

Figure 4. Structure of Module III. The fusion layers are shown in purple, while the Concat and two-layer LSTM blocks are shown in light blue.

2.3.1. Intra-Scale Time–Frequency Feature Fusion

The feature fusion stage concatenates the time-domain and frequency-domain representations at each scale. For each scale $k \in \{\mathrm{B}, \mathrm{M}, \mathrm{U}, \mathrm{P}\}$, the time-domain features $\mathbf{T}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{time}}}$ and frequency-domain features $\mathbf{F}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{freq}}}$ are integrated via a feature-level concatenation. This yields the composite feature matrix $\mathbf{C}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times (d_{\text{time}} + d_{\text{freq}})}$:

$$\mathbf{C}^{(k)} = \text{Concat}\left(\mathbf{T}^{(k)}, \mathbf{F}^{(k)}\right) \in \mathbb{R}^{L_{\text{seq}} \times (d_{\text{time}} + d_{\text{freq}})}, \tag{22}$$

where $L_{\text{seq}} = 12$ is the common sequence length.

A learnable fusion layer—composed of a linear transformation followed by a Rectified Linear Unit (ReLU) activation—is then applied to integrate cross-domain information and generate a fused feature representation:

$$\mathbf{S}^{(k)} = \text{ReLU}\left(\mathbf{C}^{(k)}\mathbf{W}_{\text{fuse}} + \boldsymbol{b}_{\text{fuse}}\right), \tag{23}$$

where $\mathbf{W}_{\text{fuse}} \in \mathbb{R}^{(d_{\text{time}} + d_{\text{freq}}) \times d_{\text{fusion}}}$ and $\boldsymbol{b}_{\text{fuse}} \in \mathbb{R}^{d_{\text{fusion}}}$ are learnable parameters, and $d_{\text{fusion}} = 64$ is the output width of the linear layer. The concatenation has $d_{\text{time}} + d_{\text{freq}} = 128$ features per position. The fusion layer yields a tensor $\mathbf{S}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{fusion}}}$ for each scale k.

2.3.2. Cross-Scale Feature Fusion

To capture the global interaction across different temporal granularities, the fused features from all four scales (Bottom, Middle, Up, and Peak) are concatenated along the feature dimension. The comprehensive multi-scale feature tensor $\mathbf{S}_{\text{total}}$ is defined as:

$$\mathbf{S}_{\text{total}} = \text{Concat}\left(\mathbf{S}^{(\mathrm{B})}, \mathbf{S}^{(\mathrm{M})}, \mathbf{S}^{(\mathrm{U})}, \mathbf{S}^{(\mathrm{P})}\right). \tag{24}$$

The resulting tensor $\mathbf{S}_{\text{total}} \in \mathbb{R}^{L_{\text{seq}} \times D_{\text{total}}}$ serves as the definitive input for the Forecasting Module, where $D_{\text{total}} = 4 \times d_{\text{fusion}}$.

2.3.3. LSTM-Based Prediction Head

The final fused feature tensor, $\mathbf{S}_{\text{total}}$, serves as the input to the LSTM-based Forecasting Module. For each sample, $\mathbf{S}_{\text{total}}$ has shape $L_{\text{seq}} \times D_{\text{total}}$, where $L_{\text{seq}} = 12$ and $D_{\text{total}} = 4 \times 64 = 256$. The two-layer LSTM produces an output sequence

$$\mathbf{H}_{\text{out}} = \text{LSTM}(\mathbf{S}_{\text{total}}) \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{LSTM}}}, \tag{25}$$

where $d_{\text{LSTM}} = 128$ is the hidden size. The output at the last sequence position is passed through a linear layer to generate all six forecast values simultaneously:

$$\widehat{y} = \mathbf{H}_{\text{out}}[L_{\text{seq}}, :]\mathbf{W}_{\text{pred}} + \boldsymbol{b}_{\text{pred}}, \tag{26}$$

where $\mathbf{W}_{\text{pred}} \in \mathbb{R}^{d_{\text{LSTM}} \times 6}$ and $\boldsymbol{b}_{\text{pred}} \in \mathbb{R}^{6}$.

2.4. Proposed Composite Training Loss

To improve the learning of multi-scale features from the time and frequency domains, a composite objective function named the COCF loss is proposed for model training. The overall loss consists of three components: an intra-scale cross-domain consistency loss, a cross-scale intra-domain orthogonality loss, and a forecasting loss for the primary prediction task.

2.4.1. Intra-Scale Cross-Domain Consistency Loss

The objective of this loss is to align the representations of the time and frequency domains within the same scale. Although processed differently, the two representations are derived from the same wind speed sequence. The loss is computed between the pooled features, as shown in Figure 5.

Let $\mathbf{T}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{time}}}$ and $\mathbf{F}^{(k)} \in \mathbb{R}^{L_{\text{seq}} \times d_{\text{freq}}}$ denote the time-domain and frequency-domain feature matrices for the k-th scale of a single sample, where $d_{\text{time}} = d_{\text{freq}} = d = 64$. To compare the overall representations at each scale, these sequences are aggregated into global context vectors by average pooling along the sequence dimension:

$$\boldsymbol{t}_k = \frac{1}{L_{\text{seq}}} \sum_{i=1}^{L_{\text{seq}}} \mathbf{T}_{i,:}^{(k)}, \tag{27}$$

$$\boldsymbol{f}_k = \frac{1}{L_{\text{seq}}} \sum_{i=1}^{L_{\text{seq}}} \mathbf{F}_{i,:}^{(k)}, \tag{28}$$

where $\boldsymbol{t}_k, \boldsymbol{f}_k \in \mathbb{R}^{d}$.

For a single sample, the consistency loss is defined as the mean absolute error (MAE) between the two pooled vectors, averaged across all K scales:

$$L_{\text{con}} = \frac{1}{Kd} \sum_{k=1}^{K} \|\boldsymbol{t}_k - \boldsymbol{f}_k\|_1, \tag{29}$$

where $\|\cdot\|_1$ sums the absolute differences over the d feature coordinates. During training, this loss is further averaged over the samples in each mini-batch.

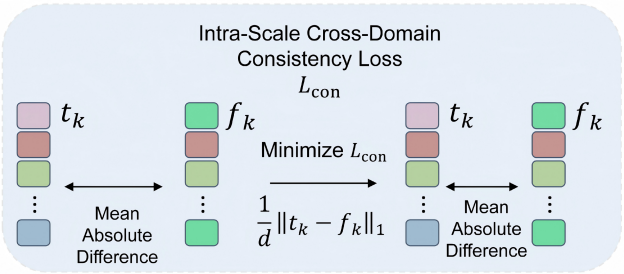

Figure 5. Intra-scale cross-domain consistency loss. Colored blocks schematically represent the components of the pooled time- and frequency-domain feature vectors.

2.4.2. Cross-Scale Intra-Domain Orthogonality Loss

To reduce redundancy across temporal scales, a cross-scale intra-domain orthogonality loss is applied to the pooled time-domain and frequency-domain representations, as illustrated in Figure 6. The loss encourages the cosine similarity of each selected scale pair to approach a prescribed target γ. The target is fixed within each training run and may differ across datasets.

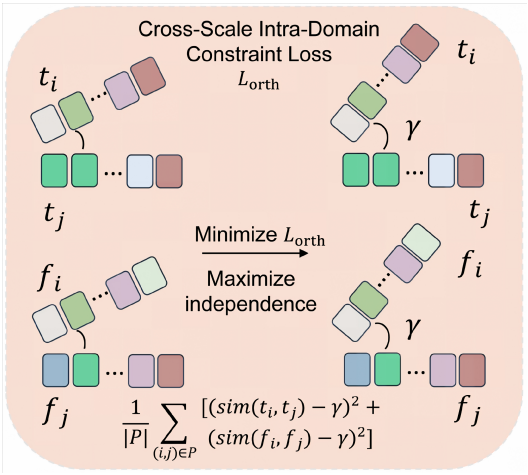

Figure 6. Cross-scale intra-domain orthogonality loss. Colored blocks schematically represent components of the pooled feature vectors at different scales in the time and frequency domains.

The loss is applied to adjacent scale pairs $P = \{(1,2), (2,3), (3,4)\}$, corresponding to (B,M), (M,U), and (U,P), respectively. These pairs connect successive resolutions in the downsampling hierarchy, so the constraint acts directly on features from neighboring scales.

For each pair $(i,j) \in P$, cosine similarity is computed separately for the time-domain and frequency-domain vectors. Given two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, their cosine similarity is

$$\mathrm{sim}(\boldsymbol{a},\boldsymbol{b}) = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|_2\,\|\boldsymbol{b}\|_2}, \tag{30}$$

where $\boldsymbol{a}\cdot\boldsymbol{b}$ denotes the dot product and $\|\cdot\|_2$ denotes the Euclidean norm.

For each scale pair, the cosine similarities are first averaged over the B samples in the mini-batch. The resulting means for the time and frequency domains are

$$\bar{c}_{ij}^{\text{time}} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{sim}(\boldsymbol{t}_{b,i},\boldsymbol{t}_{b,j}), \qquad \bar{c}_{ij}^{\text{freq}} = \frac{1}{B}\sum_{b=1}^{B}\mathrm{sim}(\boldsymbol{f}_{b,i},\boldsymbol{f}_{b,j}),$$

where b indexes the samples. The orthogonality loss is then computed as

$$L_{\text{orth}} = \frac{1}{|P|}\sum_{(i,j)\in P}\left[\left(\bar{c}_{ij}^{\text{time}}-\gamma\right)^2+\left(\bar{c}_{ij}^{\text{freq}}-\gamma\right)^2\right]. \tag{31}$$

2.4.3. Global Forecasting Loss

In this study, the primary objective of the regression task is to minimize the cumulative discrepancy between the predicted trajectories and the actual observations across the entire forecast horizon, as illustrated in Figure 7. To account for the multi-step nature of the task, MSE is employed as the global loss function. For a given training batch of N samples, the loss function L_{mse} is defined as the average squared difference across the entire forecast horizon:

$$L_{\text{mse}} = \frac{1}{N \cdot L} \sum_{i=1}^{N} \sum_{l=1}^{L} (y_{i,t+l} - \hat{y}_{i,t+l})^2, \tag{32}$$

where $y_{i,t+l}$ and $\hat{y}_{i,t+l}$ represent the ground truth and the model-generated forecast for the i-th sample at the future time step $t + l$, respectively. By minimizing L_{mse}, the model is optimized to capture the temporal dependencies and maintain point-wise accuracy across all six look-ahead steps simultaneously.

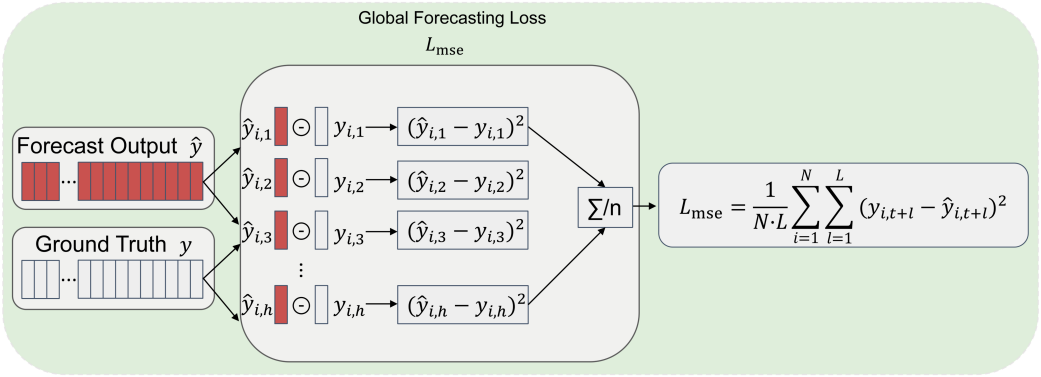

Figure 7. Schematic illustration of the global forecasting loss.

2.4.4. Total Loss Function

The total training loss is a weighted combination of the forecast loss and two regularization terms:

$$L_{\text{total}} = L_{\text{mse}} + \lambda_1 L_{\text{con}} + \lambda_2 L_{\text{orth}}, \tag{33}$$

where λ_1 and λ_2 are hyperparameters that balance the consistency loss L_{con} and the orthogonality loss L_{orth} against the primary mean squared error loss L_{mse}.

3. Implementation of COMTFFL for Wind Speed Forecasting

This section describes the wind speed forecasting workflow of COMTFFL. The overall process, illustrated in Figure 8, consists of the following three stages.

Step 1: Data Collection and Pre-processing. The original wind speed series is first divided chronologically into training, validation, and test sets in a ratio of 8:1:1. All subsets are standardized using the mean and standard deviation of the training set. Each sliding window contains 48 input values and the next six values as targets. At the start of the validation and test sets, an input window may include historical values from the preceding set. All six targets in each window fall within the corresponding split. Each input window is then downsampled by factors of 2, 4, and 8 to generate the additional scales. The Z-score normalization process is defined as:

$$\hat{x} = \frac{x - \mu}{\sigma}, \tag{34}$$

where x is the original wind speed data, $\hat{x}$ is the normalized data, μ is the mean of the training set, and σ is the standard deviation of the training set.

Step 2: Model Training with the COCF Loss. The COMTFFL model is trained using the composite COCF loss function. The consistency weight λ_1, orthogonality weight λ_2, and target cosine similarity γ are selected separately for each dataset based on validation-set performance, as described in the Implementation Details Subsection. To prevent overfitting, an early stopping strategy is adopted, and training is terminated if the validation loss does not decrease for 10 consecutive epochs.

Step 3: Forecasting with the Trained COMTFFL Model. The best-performing model is then used to generate forecasts on the test set. The predicted values are subsequently inverse-transformed to the original scale for performance evaluation.

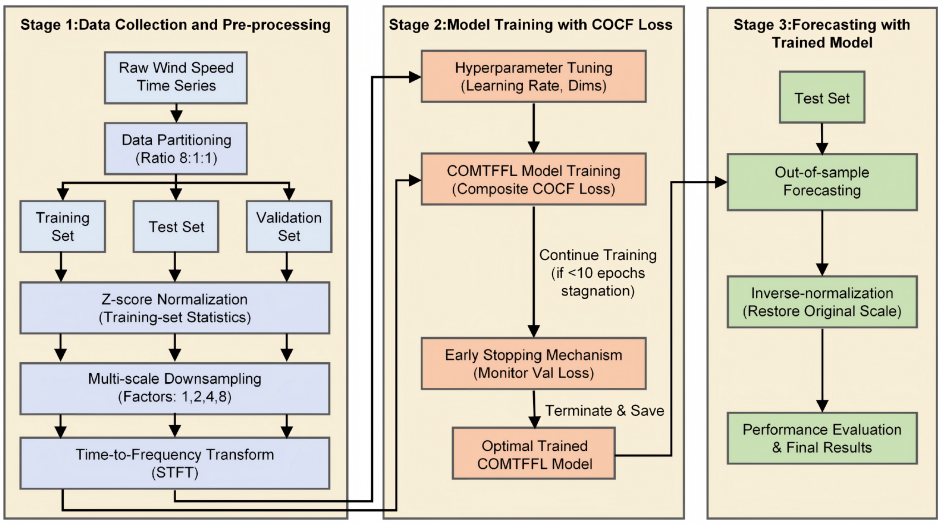

Figure 8. Flowchart of the proposed model for wind speed forecasting, COMTFFL.

4. Case Study

To evaluate the performance of COMTFFL, extensive experiments were conducted on three wind speed datasets. This section presents the datasets, experimental settings, and performance evaluation results. All experiments were performed on a workstation running Windows 10 and equipped with an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory. The model was implemented in Python 3.12 using the PyTorch 2.7.1 framework.

4.1. Data Description

To evaluate the performance of COMTFFL, three public wind speed datasets were obtained from the Southeast Asia (Original) product of the Wind Resource Database (WRDB) at https://wrdb.nlr.gov/data-viewer (accessed on 24 June 2026). The datasets contain modeled wind speeds at three sites and are referred to as Dataset A, Dataset B, and Dataset C. The site for Dataset A (site 1011267) is located at 4.2156157° N, 108.56036° E. The site for Dataset B (site 1820251) is located at 8.7215° S, 138.95633° E. The site for Dataset C (site 156598) is located at 21.174767° N, 122.93535° E. Only wind speed at 80 m above the surface, expressed in m/s, was used for forecasting. Each dataset contains 17,520 records at 30 min intervals without missing values, covering 1 January to 31 December 2021 in UTC.

For model training and evaluation, each dataset was divided into training, validation, and test subsets in a ratio of 8:1:1. A sliding-window strategy was employed for sample generation, where the previous 24 h of wind speed values were used to forecast the next 3 h. Formally, at time t, the input and output sequences are denoted by $\{y_{t-47}, y_{t-46}, y_{t-45}, y_{t-44}, \ldots, y_{t-1}, y_t\}$ and $\{y_{t+1}, \ldots, y_{t+5}, y_{t+6}\}$, respectively. A single model is trained to generate the forecasts for all future time steps simultaneously. Table 1 summarizes the key statistical properties of the three datasets.

Table 1. Information on the three datasets.

Dataset	Resolution	Number	Average	Variance	Minimum	Maximum	Skewness	Kurtosis
A	30 min	17,520	3.9177	5.4287	0.04	12.94	0.1362	0.7303
B	30 min	17,520	5.7866	6.9038	0.05	13.77	0.0040	−0.7163
C	30 min	17,520	7.3302	10.7121	0.05	23.62	0.5123	0.1771

4.2. Evaluation Metrics

To evaluate the forecasting performance of different models, five standard metrics are employed: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), the coefficient of determination (R^2), and the Index of Agreement (IA). Their mathematical formulations are given as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|F_i - O_i|, \tag{35}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(F_i - O_i)^2}, \tag{36}$$

$$\mathrm{SMAPE}(\%) = \frac{100}{n}\sum_{i=1}^{n}\frac{|F_i - O_i|}{|F_i| + |O_i| + \epsilon}, \qquad \epsilon = 10^{-8}, \tag{37}$$

$$\mathrm{IA} = 1 - \frac{\sum_{i=1}^{n}(F_i - O_i)^2}{\sum_{i=1}^{n}\left(|F_i - \overline{O}| + |O_i - \overline{O}|\right)^2}, \tag{38}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(F_i - O_i)^2}{\sum_{i=1}^{n}(O_i - \overline{O})^2}, \tag{39}$$

where n is the number of forecast–observation pairs used in the calculation, F_i is the i-th forecast, O_i is the corresponding observed value, and $\overline{O} = n^{-1}\sum_{i=1}^{n} O_i$ is the mean of the observed values. Lower MAE, RMSE, and SMAPE values indicate smaller forecast errors, while higher R^2 and IA values indicate better agreement with the observed values.

4.3. Implementation Details

To validate the forecasting performance of the proposed model, seven widely used benchmark architectures for wind speed forecasting were selected for comparison, including iTransformer [21], Autoformer [39], TimesNet [20], TimeXer [40], Transformer [41], Informer [42], and FEDformer [43]. The benchmark implementations are available at https://github.com/thuml/Time-Series-Library (accessed on 28 July 2026). All models were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32 for a maximum of 100 epochs (Table 2). The search space for hyperparameters in COMTFFL and baseline models is shown in Table 3. The final hyperparameter settings are listed in Table A1. The model parameter counts are reported in Table A2. In addition, an early stopping strategy was applied to all models, and training was terminated if the validation loss did not decrease for 10 consecutive epochs.

Table 2. Model training settings.

Model Training Setup	Values
Batch size	32
Epoch	100
Optimizer	Adam
Learning rate	0.001
Patience	10

For COMTFFL, the consistency weight λ_1, the orthogonality weight λ_2, and the target cosine similarity γ were selected separately for each dataset by comparing candidate configurations on the validation set. The selected configurations were then evaluated on the test set. In addition, ablation studies were conducted to evaluate the contributions of

the Multi-Scale Feature Extraction Module, the Dual-Domain Feature Fusion Module, and the proposed COCF loss.

Table 3. Hyperparameter search space for COMTFFL and baseline models.

Hyperparameters	Search Space
Number of encoder layers	{1, 2, 3}
Number of decoder layers	{1, 2}
Number of attention heads	{4, 8, 16}
Feature dimension	{32, 64, 128, 256, 512}
Dimension of FFN	{32, 512, 1024, 2048}
Dropout rate	{0.05, 0.1, 0.2}
Patch length (for TimeXer)	{8, 16, 24}
Top k periods (for TimesNet)	{1, 3, 5}
Version (for FEDformer)	{Fourier, Wavelet}
Downsampling ratio (for COMTFFL)	{2, 4, 8}
Consistency weight λ_1 (for COMTFFL)	[0.01, 0.2] with a step of 0.01
Orth weight λ_2 (for COMTFFL)	[0.01, 0.2] with a step of 0.01

4.4. Experimental Results and Comparative Analysis

In this section, the forecasting performance of COMTFFL is evaluated against seven state-of-the-art (SOTA) models on three wind speed datasets. Table 4 presents the forecasting results of all models, and Figure 9 further illustrates their relative performance across different evaluation metrics.

Table 4. Average wind speed forecasting results on the three datasets.

Dataset	Model	RMSE	MAE	SMAPE	R^2	IA
Dataset A	iTransformer	0.7239	0.5403	7.1658	0.8768	0.9681
	Autoformer	0.9296	0.6833	8.4881	0.8024	0.9459
	TimesNet	0.6684	0.4736	6.0872	0.8979	0.9733
	TimeXer	0.6527	0.4543	5.8808	0.8980	0.9728
	Transformer	0.6843	0.4859	6.0848	0.8929	0.9732
	Informer	1.0531	0.7993	9.7506	0.7464	0.9270
	FEDformer	0.6757	0.4931	6.4929	0.8956	0.9722
	COMTFFL	0.5979	0.4538	5.7813	0.9050	0.9748
Dataset B	iTransformer	0.7522	0.5490	11.0327	0.9144	0.9781
	Autoformer	1.1485	0.8582	16.1537	0.8003	0.9466
	TimesNet	0.7275	0.5223	10.5813	0.9199	0.9796
	TimeXer	0.7133	0.5175	10.3224	0.9203	0.9794
	Transformer	0.7130	0.5223	10.4352	0.9163	0.9799
	Informer	0.7405	0.5445	10.4885	0.9170	0.9783
	FEDformer	0.7894	0.5803	11.8934	0.9057	0.9754
	COMTFFL	0.6768	0.5145	9.8457	0.9231	0.9795
Dataset C	iTransformer	0.9929	0.7358	4.4257	0.8861	0.9705
	Autoformer	1.3304	0.9625	5.4731	0.7955	0.9442
	TimesNet	0.7631	0.5392	3.2756	0.9327	0.9828
	TimeXer	0.6893	0.4742	2.8796	0.9451	0.9860
	Transformer	0.7157	0.4929	3.0233	0.9408	0.9850
	Informer	0.9140	0.6355	3.7166	0.9034	0.9771
	FEDformer	0.7460	0.5454	3.2522	0.9357	0.9830
	COMTFFL	0.6275	0.4707	2.8304	0.9455	0.9860

Table 4 summarizes the forecasting performance of the compared models on the three datasets. Figure 9 illustrates the differences across evaluation metrics. On Dataset C,

COMTFFL achieves an RMSE of 0.6275, compared with 0.6893 for TimeXer. COMTFFL's MAE is lower than TimeXer's by 0.0005 and 0.0035 m/s on Datasets A and C, respectively.

Table 5 compares the MAE of COMTFFL and TimeXer across two runs on Datasets A and C. COMTFFL achieved a slightly lower MAE in the first run on both datasets, while TimeXer achieved a lower MAE in the second run. The two models had nearly identical MAE on Dataset A, and TimeXer had the lowest MAE on Dataset C.

Table 5. MAE of COMTFFL and TimeXer on Datasets A and C across two runs.

Dataset	Model	Run 1	Run 2	Mean ± SD
Dataset A	COMTFFL	0.4538	0.4660	0.4599 ± 0.0086
	TimeXer	0.4543	0.4654	0.4598 ± 0.0078
Dataset C	COMTFFL	0.4707	0.4852	0.4780 ± 0.0103
	TimeXer	0.4742	0.4662	0.4702 ± 0.0057

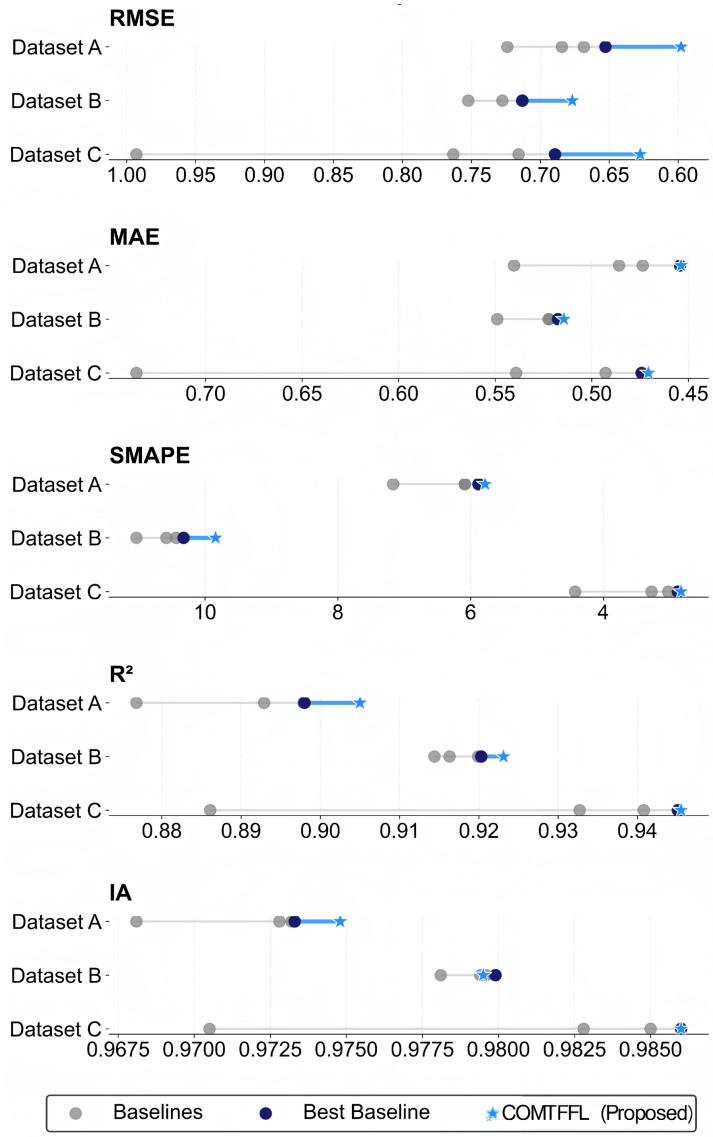

Figure 9. Comparison of forecasting performance between COMTFFL and benchmark models.

4.5. Step-Wise Error Analysis and Stability Evaluation

Figure 10 presents the step-wise forecasting results of different models. For all three datasets, forecasting errors increase as the forecasting horizon extends. Although several

benchmark models achieve lower errors at the first forecasting step, COMTFFL maintains a slower increase in error and achieves better performance at later forecasting steps.

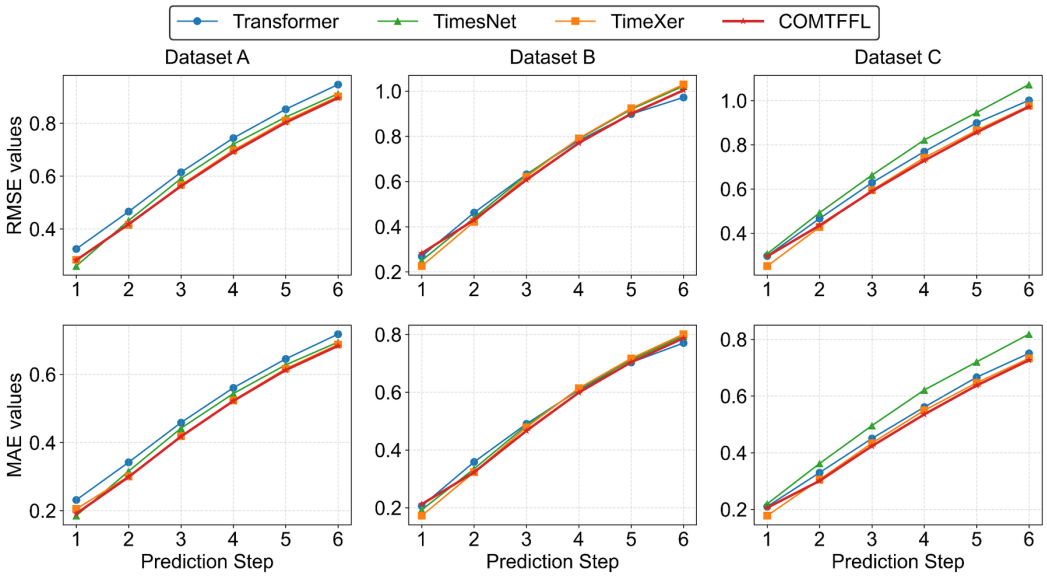

Figure 10. Step-wise RMSE and MAE on different forecasting horizons.

For Datasets A and B, TimesNet achieves the lowest RMSE at the first forecasting step. However, COMTFFL performs better at later forecasting steps and achieves the lowest RMSE at Step 6. A similar trend is observed on Dataset C. While TimeXer performs slightly better at the shortest forecasting horizon, COMTFFL exhibits superior performance as the forecasting horizon extends. In contrast, the forecasting errors of Autoformer and Informer increase more rapidly with the forecasting horizon.

Overall, COMTFFL achieves better performance at longer forecasting steps. This may be attributed to the multi-scale temporal–frequency feature fusion framework, which enables the model to capture both local variations and broader temporal patterns from wind speed series.

4.6. Analysis of Forecasting Trajectories and Dynamic Tracking

Figure 11 compares the first-step forecasts of COMTFFL and four benchmark models over selected test segments from the three datasets. On Dataset A, the COMTFFL forecasts follow several observed peaks and troughs more closely than those of some benchmark models. On Dataset B, COMTFFL follows the short-term fluctuations in the displayed segment. On Dataset C, COMTFFL tracks the rapid increase around time step 240, while some benchmark forecasts lag behind the observed increase. Overall forecasting performance is summarized in Table 4.

4.7. Effect of STFT Framing

The original STFT setting, with a 64-sample window and a hop of 16 samples, was compared with a short-window setting. At scale factors $1, 2, 4, 8$, the short-window setting uses window lengths of $32, 16, 8, 4$ and hop lengths of $8, 4, 2, 1$, respectively. It extracts three complete frames per scale without centering or reflection padding. The short frames are zero-padded to 64 points for the FFT. Both settings therefore yield 33 one-sided magnitude bins per frame. For each dataset, the two runs used the same model architecture, data split, loss parameters, and random seed (42).

Table 6 includes the main experimental results for reference. Boldface marks the lower error between the two matched runs for each dataset. The original framing yielded lower

RMSE and MAE on all three datasets. With short-window framing, RMSE increased by 20.2%, 25.2%, and 18.4% on Datasets A, B, and C, respectively, and MAE increased by 26.3%, 34.2%, and 24.0%. These differences reflect the combined changes in window length, hop length, and boundary handling.

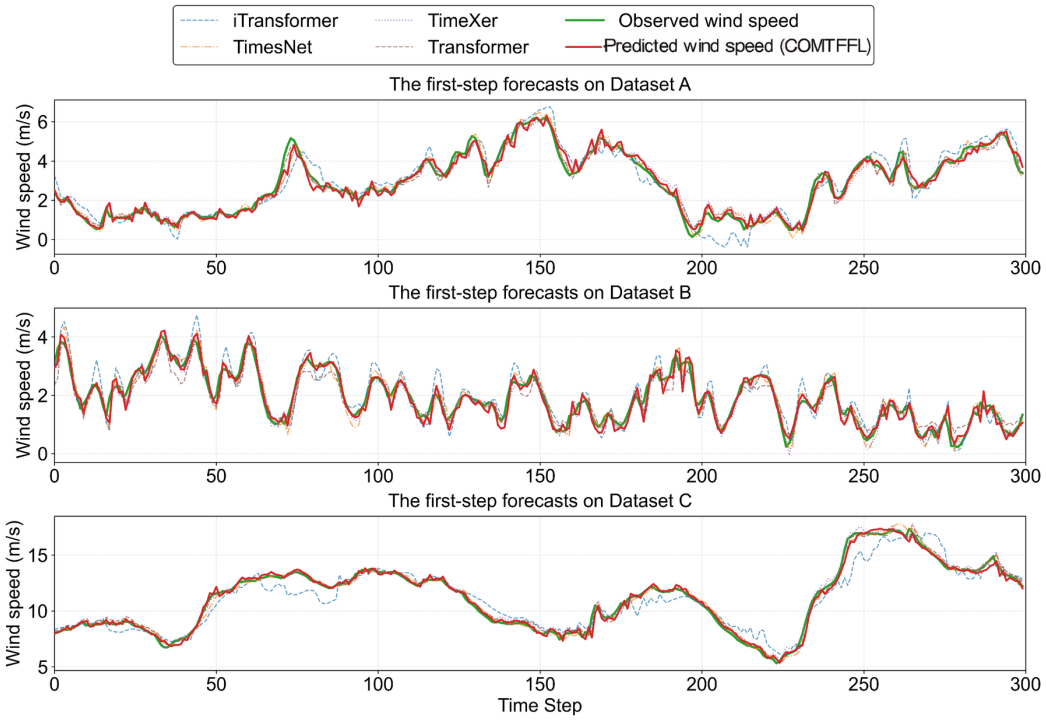

Figure 11. Comparison of wind speed forecasts and observed values.

Table 6. Forecast errors for the STFT framing comparison. Bold values indicate the lower errors between the two framing settings for each dataset, excluding the reference results.

Dataset	Configuration	RMSE	MAE
Dataset A	Main experiment (reference)	0.5979	0.4538
	Original framing	**0.6068**	**0.4616**
	Short-window framing	0.7294	0.5828
Dataset B	Main experiment (reference)	0.6768	0.5145
	Original framing	**0.6861**	**0.5269**
	Short-window framing	0.8589	0.7071
Dataset C	Main experiment (reference)	0.6275	0.4707
	Original framing	**0.6415**	**0.4822**
	Short-window framing	0.7598	0.5981

4.8. Effects of Multi-Scale and Dual-Domain Processing

Table 7 compares four architectural variants trained using only the forecast MSE loss. COMTFFL-Base retains the four-scale, dual-domain architecture of the full COMTFFL model. The full model additionally uses the consistency and orthogonality losses. The other three variants are derived from COMTFFL-Base. The single-scale dual-domain variant retains both domains at the original resolution. The multi-scale time-only and frequency-only variants retain all four scales and use only time-domain and frequency-domain features, respectively.

COMTFFL-Base achieved lower RMSE and MAE than both four-scale single-domain variants on all three datasets. Among the dual-domain variants, COMTFFL-Base achieved a slightly lower RMSE on Dataset A, while the single-scale variant achieved a slightly lower

MAE. On Datasets B and C, the single-scale variant achieved lower RMSE and MAE. Thus, the benefit of adding coarser scales varied by dataset and metric in this experiment.

Table 7. Forecasting performance of the four model variants. Bold values indicate the best result for each metric within each dataset.

Dataset	Variant	RMSE	MAE	SMAPE (%)	R^2	IA
Dataset A	COMTFFL-Base	**0.6079**	0.4680	**5.8870**	**0.9014**	**0.9735**
	Single-scale dual-domain	0.6111	**0.4668**	5.9168	0.8995	0.9732
	Multi-scale time-only	0.7309	0.5836	7.2423	0.8582	0.9608
	Multi-scale frequency-only	1.8713	1.6541	19.8263	−0.1299	0.6577
Dataset B	COMTFFL-Base	0.6825	0.5219	9.7924	0.9210	0.9780
	Single-scale dual-domain	**0.6792**	**0.5155**	**9.7280**	**0.9221**	**0.9788**
	Multi-scale time-only	0.8423	0.6835	12.7306	0.8797	0.9651
	Multi-scale frequency-only	3.1714	3.0317	36.2780	−0.7500	0.4601
Dataset C	COMTFFL-Base	0.6405	0.4859	2.8917	0.9435	0.9849
	Single-scale dual-domain	**0.6358**	**0.4780**	**2.8566**	**0.9445**	**0.9858**
	Multi-scale time-only	0.7966	0.6328	3.7630	0.9103	0.9764
	Multi-scale frequency-only	2.9268	2.6169	16.5203	−0.3466	0.6738

4.9. Effectiveness of Consistency and Orthogonality Constraints

The consistency loss (L_{con}) and orthogonality loss (L_{orth}) address limitations of intra-scale cross-domain feature inconsistency and cross-scale intra-domain feature redundancy. To evaluate their contribution, three variants are compared with the full COMTFFL model. Specifically, COMTFFL-Con retains only L_{con}, COMTFFL-Ort retains only L_{orth}, and COMTFFL-Base removes both constraints. Table 8 and Figure 12 present the forecasting results across all evaluation metrics. The loss ablation was run separately from the structural ablation in Table 7.

Table 8. Ablation study of the consistency and orthogonality losses.

Data	Model	RMSE	MAE	SMAPE	R^2	IA
Dataset A	COMTFFL-Base	0.6137	0.4674	5.9202	0.9009	0.9739
	COMTFFL-Con	0.6043	0.4576	5.8111	0.9031	0.9742
	COMTFFL-Ort	0.6059	0.4612	5.8568	0.9028	0.9736
	COMTFFL	0.5979	0.4538	5.7813	0.9050	0.9748
Dataset B	COMTFFL-Base	0.6972	0.5316	9.9202	0.9175	0.9771
	COMTFFL-Con	0.6913	0.5268	9.8646	0.9199	0.9779
	COMTFFL-Ort	0.6908	0.5245	9.8813	0.9204	0.9783
	COMTFFL	0.6768	0.5145	9.8457	0.9231	0.9795
Dataset C	COMTFFL-Base	0.6542	0.5013	2.9864	0.9419	0.9853
	COMTFFL-Con	0.6329	0.4772	2.8832	0.9450	0.9858
	COMTFFL-Ort	0.6366	0.4804	2.9280	0.9443	0.9854
	COMTFFL	0.6275	0.4707	2.8304	0.9455	0.9860

On Dataset C, adding the consistency loss reduces RMSE from 0.6542 to 0.6329, MAE from 0.5013 to 0.4772, and SMAPE from 2.9864% to 2.8832%. COMTFFL-Ort also yields lower RMSE and MAE than COMTFFL-Base, with values of 0.6366 and 0.4804, respectively. Its SMAPE also decreases to 2.9280%. Both single-loss variants achieve higher R^2 and IA than COMTFFL-Base on Dataset C.

The full COMTFFL model achieves the lowest RMSE among the four loss settings on each dataset in Table 8. On Dataset C, the RMSE decreases from 0.6542 for COMTFFL-Base to 0.6275 for the full model, a reduction of approximately 4.1%. The full model achieves an MAE of 0.4707 and an SMAPE of 2.8304%.

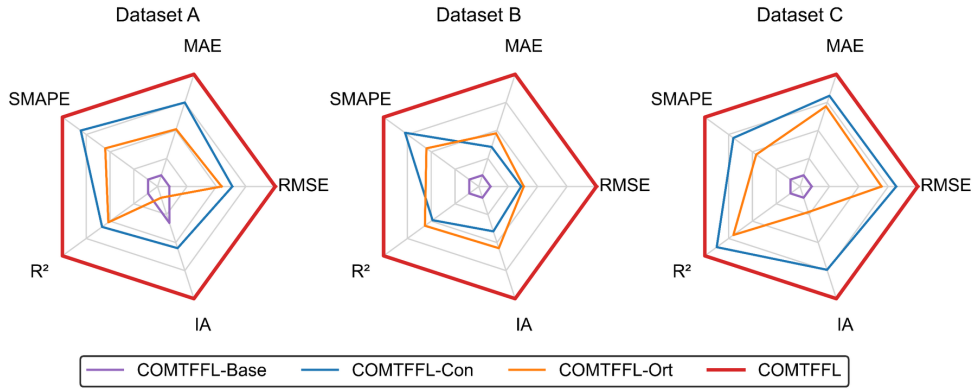

Figure 12. Ablation study results for the consistency and orthogonality losses.

4.10. Forecasting Performance of Ablation Models Across Different Prediction Horizons

Table 8 presents the overall forecasting results of different variants. To further examine their performance at different forecasting horizons, the step-wise RMSE and MAE results are presented in Figure 13.

For all three datasets, forecasting errors increase as the prediction horizon extends. COMTFFL-Con and COMTFFL-Ort yield lower errors than COMTFFL-Base at most forecasting steps. On Dataset B, the RMSE of COMTFFL-Base increases from 0.3051 at Step 1 to 1.0233 at Step 6, while the variants with regularization show a smaller increase over the same horizon.

The best results are achieved by the full COMTFFL model. Compared with COMTFFL-Base, lower RMSE and MAE values are obtained at the final forecasting step across all three datasets. For example, the RMSE decreases from 0.9122 to 0.8956 on Dataset A and from 0.9905 to 0.9716 on Dataset C. The results suggest that the two losses provide complementary benefits. By jointly enforcing intra-scale cross-domain feature consistency and cross-scale intra-domain feature orthogonality, the full model maintains more informative feature representations and achieves better forecasting performance at longer prediction horizons.

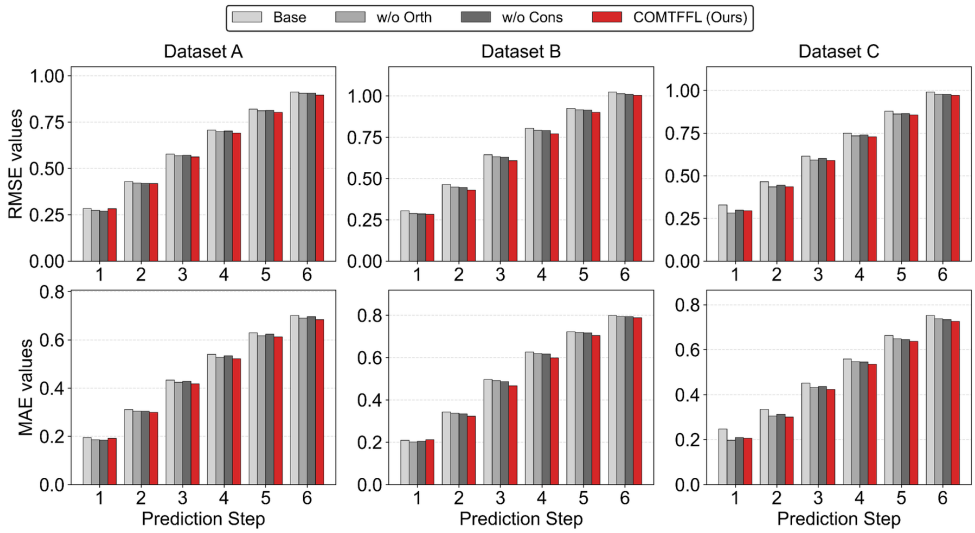

Figure 13. Ablation study on multi-step forecasting performance.

4.11. Cross-Domain Feature Consistency Analysis

Table 9 presents the mean cross-domain L_2 distances for COMTFFL-Base and full COMTFFL. At each scale, the distance is computed between the flattened time- and frequency-domain features of each test window and averaged across all test windows.

Full COMTFFL yields a lower mean distance in all 12 dataset–scale combinations. The reduction is particularly evident at Scale 4. On Dataset A, the distance decreases from 6.7699 to 0.0185, while the corresponding distances for Datasets B and C decrease to 0.0349 and 0.0035, respectively.

The distances also vary across datasets and scales. For example, at Scale 3 on Dataset C, the distance decreases from 23.6772 to 0.0081. These results indicate closer alignment between paired time- and frequency-domain representations when the consistency and orthogonality losses are applied together.

To further examine cross-domain alignment, Figure 14 presents PCA visualizations of the time- and frequency-domain features at Scale 1. The upper row shows the features from COMTFFL-Base, and the lower row shows those from full COMTFFL. For each dataset, PCA was fitted jointly to the features from both models and both domains. The two rows use the same projection and axis limits. Each line connects the time- and frequency-domain features of the same test window.

Table 9. Mean cross-domain L_2 distances for COMTFFL-Base and full COMTFFL.

Scale	Dataset	COMTFFL-Base	Full COMTFFL
1	A	31.5925	14.7738
1	B	31.4188	14.2949
1	C	29.0806	10.2442
2	A	28.1244	0.6478
2	B	25.2857	4.6019
2	C	27.3694	1.6426
3	A	26.7717	0.5472
3	B	22.3681	5.0606
3	C	23.6772	0.0081
4	A	6.7699	0.0185
4	B	23.6115	0.0349
4	C	4.6957	0.0035

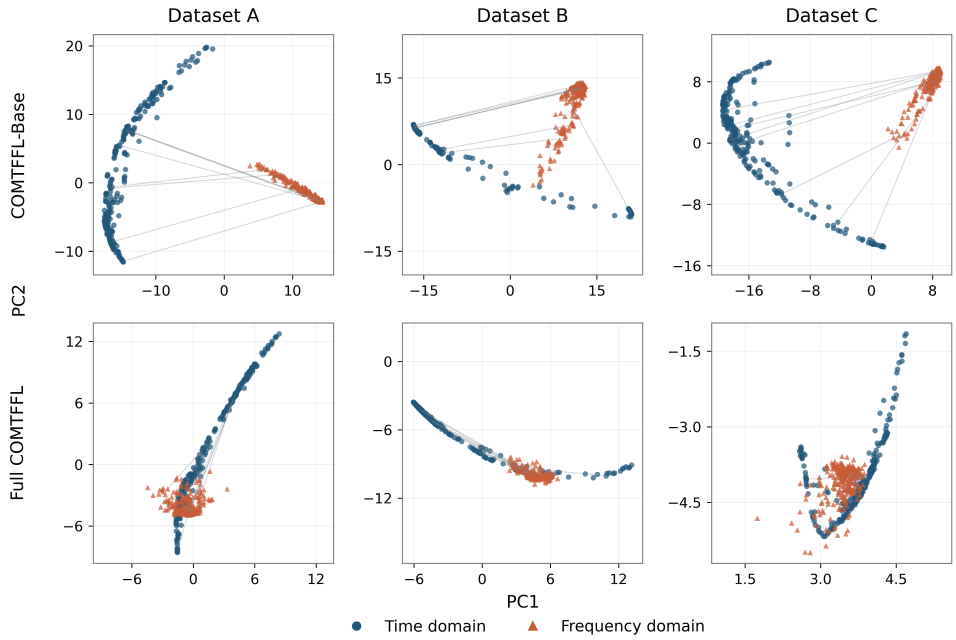

Figure 14. Shared PCA projections of Scale 1 time- and frequency-domain features for COMTFFL-Base (**top**) and full COMTFFL (**bottom**). Gray lines connect paired time- and frequency-domain features from the same test window.

As shown in Figure 14, the displayed pairs tend to be closer in the projected space for full COMTFFL, particularly on Datasets B and C. This pattern is consistent with the lower mean L_2 distances at Scale 1 reported in Table 9. Together, the table and the visualization show closer cross-domain alignment under the combined consistency and orthogonality constraints.

4.12. Cross-Scale Orthogonality Constraint Representation

Figure 15 compares the cross-scale cosine similarities of COMTFFL-Base and COMTFFL-Ort in the time and frequency domains. Both models were trained with forecast MSE, and COMTFFL-Ort additionally used the orthogonality loss. In COMTFFL-Base, the time-domain similarity reaches 0.86 for the scale pair $(2,3)$ on Dataset A and 0.85 for the scale pair $(1,3)$ on Dataset B. On Dataset A, the frequency-domain similarities for the scale pairs $(2,4)$ and $(3,4)$ both reach 1.00 at the reported precision.

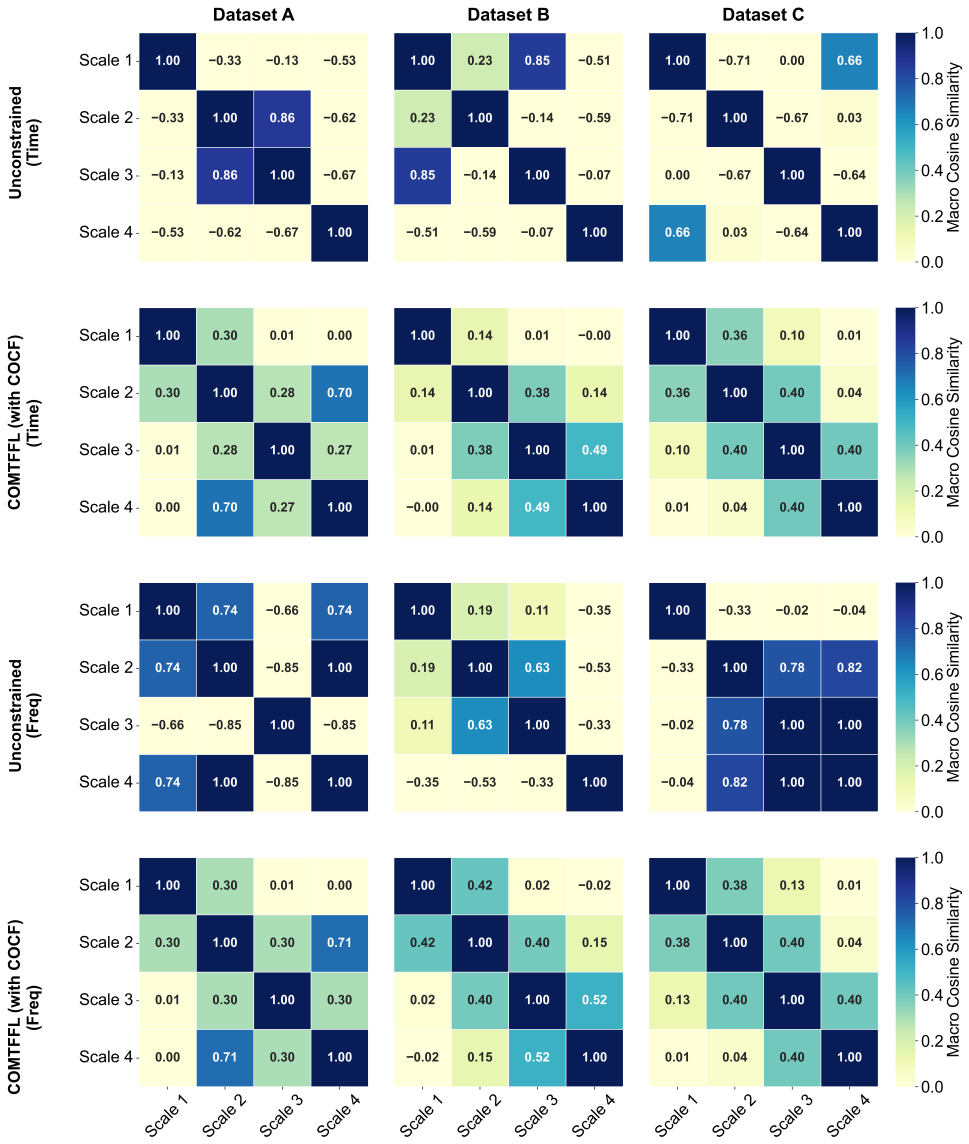

Figure 15. Cosine similarity matrices of multi-scale features before and after applying the orthogonality loss.

The orthogonality loss directly constrains the adjacent scale pairs $(1,2)$, $(2,3)$, and $(3,4)$. For each dataset, the target cosine similarity γ provides a reference for interpreting the similarities of these pairs. The matrices also show similarities for non-adjacent pairs.

5. Conclusions

To address multi-scale feature redundancy and cross-domain noise interference in wind speed forecasting, this study proposed COMTFFL, a dual-branch model that extracts and fuses multi-scale time- and frequency-domain features. The COCF loss encourages cross-domain feature alignment and regularizes cross-scale feature similarity. Compared with COMTFFL-Base, the full model yielded smaller cross-domain L_2 distances at all four scales on all three datasets. The orthogonality-only variant showed changes in cross-scale cosine similarity in both domains. COMTFFL achieved the lowest RMSE on all three datasets in the main comparison. Its MAE ranking relative to TimeXer reversed between the two runs on Datasets A and C. In practice, the proposed framework may support wind speed forecasting for power dispatch.

Author Contributions: Y.W.: Conceptualization, Funding acquisition, Supervision, and Writing—review and editing; W.L.: Data curation, Methodology, Visualization, and Writing—original draft; F.Z.: Data curation, Methodology, Supervision, and Writing—review and editing. 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. 62376289) and the Natural Science Foundation of Hunan Province, China (Grant No. 2024JJ4069).

Data Availability Statement: The datasets analyzed in this study are available through the Wind Resource Database (WRDB) Data Viewer at https://wrdb.nlr.gov/data-viewer (accessed on 24 June 2026). This study uses the Southeast Asia (Original) product, 2021 UTC exports for sites 1011267, 1820251, and 156598, and the wind speed variable at 80 m above the surface sampled at 30 min intervals. Product documentation is available at https://developer.nlr.gov/docs/wind/wind-toolkit/southeast-asia-wind-download/ (accessed on 24 June 2026).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Model Configurations and Computational Cost

Table A1. Hyperparameter settings for the six-step forecasting experiments.

Model	Width	FFN Width	Enc. Layers	Heads	Dropout	Max. Epochs
iTransformer	512	512	3	8	0.10	100
Autoformer	512	2048	2	8	0.10	100
TimesNet	32	32	2	8	0.10	100
TimeXer	256	512	1	8	0.10	100
Transformer	512	2048	2	8	0.10	100
Informer	512	2048	2	8	0.10	100
FEDformer	512	2048	2	8	0.10	100
COMTFFL	64	256/2048	2	4	0.10	100
Model	Model-specific settings					
Autoformer	One decoder layer, moving-average window of 25.					
TimesNet	Top-$k = 5$, with 6 Inception kernels.					
TimeXer	Patch length of 16.					
Transformer	One decoder layer.					
Informer	One decoder layer, attention factor of 3.					
FEDformer	One decoder layer, Fourier variant with 32 configured modes.					
COMTFFL	Scale factors 1, 2, 4, 8; fusion width of 64; and a two-layer LSTM with hidden size of 128. The time and frequency encoders have FFN widths of 256 and 2048, respectively. The original STFT uses a window length of 64 and a hop length of 16.					

Table A1. *Cont.*

Model	Width	FFN Width	Enc. Layers	Heads	Dropout	Max. Epochs
COMTFFL (loss settings)	The settings $(\lambda_1, \lambda_2, \gamma)$ give the consistency weight, orthogonality weight, and target cosine similarity, respectively. Dataset A $(0.04, 0.06, 0.30)$ Dataset B $(0.08, 0.01, 0.40)$ Dataset C $(0.06, 0.12, 0.40)$					

Table A2. Model parameter counts.

Model	Parameters (M)
iTransformer	4.763
Autoformer	10.505
TimesNet	1.191
TimeXer	0.814
Transformer	6.309
Informer	11.306
FEDformer	14.044
COMTFFL	3.055
COMTFFL-Base	3.055
Single-scale dual-domain	0.905
Multi-scale time-only	0.748
Multi-scale frequency-only	2.605

References

1. Ullah, S.; Chen, X.; Han, H.; Wu, J.; Dong, J.; Liu, R.; Ding, W.; Liu, M.; Li, Q.; Qi, H.; et al. A novel hybrid ensemble approach for wind speed forecasting with dual-stage decomposition strategy using optimized GRU and transformer models. *Energy* **2025**, *329*, 136739. https://doi.org/10.1016/j.energy.2025.136739.
2. International Renewable Energy Agency; International Labour Organization. Renewable Energy and Jobs: Annual Review 2025. 2026. Available online: https://www.irena.org/Publications/2026/Jan/Renewable-energy-and-jobs-Annual-review-2025 (accessed on 5 August 2026).
3. Saini, V.K.; Kumar, R.; Al-Sumaiti, A.S.; Sujil, A.; Heydarian-Forushani, E. Learning based short term wind speed forecasting models for smart grid applications: An extensive review and case study. *Electr. Power Syst. Res.* **2023**, *222*, 109502. https://doi.org/10.1016/j.epsr.2023.109502.
4. Wang, Y.; Zhang, F.; Kou, H.; Zou, R.; Hu, Q.; Wang, J.; Srinivasan, D. A review of predictive uncertainty modeling techniques and evaluation metrics in probabilistic wind speed and wind power forecasting. *Appl. Energy* **2025**, *396*, 126234. https://doi.org/10.1016/j.apenergy.2025.126234.
5. Wang, Y.; Duan, X.; Zhang, F.; Wu, G.; Zou, R.; Wan, J.; Hu, Q. MFFDM-WLS: A multi-granularity feature-based coherent forecasting method for temporal hierarchical wind speed time series. *Appl. Energy* **2025**, *400*, 126615. https://doi.org/10.1016/j.apenergy.2025.126615.
6. Liu, Z.; Guo, H.; Zhang, Y.; Zuo, Z. A comprehensive review of wind power prediction based on machine learning: Models, applications, and challenges. *Energies* **2025**, *18*, 350. https://doi.org/10.3390/en18020350.
7. Lydia, M.; Kumar, S.S.; Selvakumar, A.I.; Kumar, G.E.P. Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Convers. Manag.* **2016**, *112*, 115–124. https://doi.org/10.1016/j.enconman.2016.01.007.
8. Aasim; Singh, S.N.; Mohapatra, A. Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting. *Renew. Energy* **2019**, *136*, 758–768. https://doi.org/10.1016/j.renene.2019.01.031.
9. Zhou, S.; Xie, G.; Wang, Y.; Wang, Y. Adaptive Full-Spectrum Dynamics Modelling for Wind Power Forecasting via Heterogeneous Cascaded Experts. *Appl. Soft Comput.* **2026**, *204*, 116476. https://doi.org/10.1016/j.asoc.2026.116476.
10. Malik, P.; Gehlot, A.; Singh, R.; Gupta, L.R.; Thakur, A.K. A review on ANN based model for solar radiation and wind speed prediction with real-time data. *Arch. Comput. Methods Eng.* **2022**, *29*, 3183–3201. https://doi.org/10.1007/s11831-021-09687-3.
11. Sarp, A.O.; Mengüç, E.C.; Peker, M.; Güvenç, B.Ç. Data-adaptive censoring for short-term wind speed predictors based on MLP, RNN, and SVM. *IEEE Syst. J.* **2022**, *16*, 3625–3634. https://doi.org/10.1109/JSYST.2022.3150749.

12. Inan, T.; Baba, A.F. Prediction of wind speed using artificial neural networks and ANFIS methods (observation buoy example). In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*; IEEE: New York, NY, USA, 2020; pp. 1–5. https://doi.org/10.1109/ASYU50717.2020.9259894.
13. Cai, C.; Zhang, L.; Zhou, J. DMPR: A novel wind speed forecasting model based on optimized decomposition, multi-objective feature selection, and patch-based RNN. *Energy* **2024**, *310*, 133277. https://doi.org/10.1016/j.energy.2024.133277.
14. Li, J.; Song, Z.; Wang, X.; Wang, Y.; Jia, Y. A novel offshore wind farm typhoon wind speed prediction model based on PSO–Bi-LSTM improved by VMD. *Energy* **2022**, *251*, 123848. https://doi.org/10.1016/j.energy.2022.123848.
15. Shen, Z.; Fan, X.; Zhang, L.; Yu, H. Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network. *Ocean Eng.* **2022**, *254*, 111352. https://doi.org/10.1016/j.oceaneng.2022.111352.
16. Song, J.; Peng, X.; Yang, Z.; Wei, P.; Wang, B.; Wang, Z. A novel wind power prediction approach for extreme wind conditions based on TCN-LSTM and transfer learning. In *2022 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*; IEEE: New York, NY, USA, 2022; pp. 1410–1415. https://doi.org/10.1109/ICPSAsia55496.2022.9949665.
17. Cai, X.; Li, D.; Zou, Y.; Liu, Z.; Heidari, A.A.; Chen, H. A hybrid wind speed forecasting model with rolling mapping decomposition and temporal convolutional networks. *Energy* **2025**, *324*, 135673. https://doi.org/10.1016/j.energy.2025.135673.
18. Wu, H.; Meng, K.; Fan, D.; Zhang, Z.; Liu, Q. Multistep short-term wind speed forecasting using transformer. *Energy* **2022**, *261*, 125231. https://doi.org/10.1016/j.energy.2022.125231.
19. Bentsen, L.Ø.; Warakagoda, N.D.; Stenbro, R.; Engelstad, P. Spatio-temporal wind speed forecasting using graph networks and novel Transformer architectures. *Appl. Energy* **2023**, *333*, 120565. https://doi.org/10.1016/j.apenergy.2022.120565.
20. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-variation modeling for general time series analysis. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023. Available online: https://openreview.net/forum?id=ju_Uqw384Oq (accessed on 5 August 2026).
21. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. ITransformer: Inverted transformers are effective for time series forecasting. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024. Available online: https://openreview.net/forum?id=JePfAI8fah (accessed on 5 August 2026).
22. Xu, Z.; Zeng, A.; Xu, Q. FITS: Modeling time series with 10k parameters. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024. Available online: https://openreview.net/forum?id=m2getD1hpk (accessed on 5 August 2026).
23. Zhu, L.; Xu, W.; Zhu, C.; Gao, J.; Mi, L.; Deng, F.; Qu, J. AFMT: Adaptive frequency decomposition and multi-scale transformer for time series forecasting. *Inf. Sci.* **2026**, *726*, 122735. https://doi.org/10.1016/j.ins.2025.122735.
24. Liu, C.; He, S.; Liu, H.; Chen, J.; Dong, H. WindTrans: Transformer-based wind speed forecasting method for high-speed railway. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 4947–4963. https://doi.org/10.1109/TITS.2023.3337150.
25. Wang, Y.; Xu, H.; Huang, Y.; Zhang, F.; Kou, H.; Zou, R.; Hu, Q.; Srinivasan, D. An End-to-End Ensemble Learning Approach for Enhancing Wind Power Forecasting. *IEEE Trans. Sustain. Energy* **2026**, *17*, 435–448. https://doi.org/10.1109/TSTE.2025.3582425.
26. Lv, L.; Wu, Z.; Zhang, J.; Zhang, L.; Tan, Z.; Tian, Z. A VMD and LSTM based hybrid model of load forecasting for power grid security. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6474–6482. https://doi.org/10.1109/TII.2021.3130237.
27. Yang, Z.; Che, J. A two stage feature extraction and synchronized feature-parameter learning framework for reliable multistep wind speed forecasting. *Energy* **2025**, *340*, 139349. https://doi.org/10.1016/j.energy.2025.139349.
28. Han, Y.; Mi, L.; Shen, L.; Cai, C.S.; Liu, Y.; Li, K.; Xu, G. A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting. *Appl. Energy* **2022**, *312*, 118777. https://doi.org/10.1016/j.apenergy.2022.118777.
29. Ibrahim, A.; Mirjalili, S.; El-Said, M.; Ghoneim, S.S.M.; Al-Harthi, M.M.; Ibrahim, T.F.; El-Kenawy, E.-S.M. Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm. *IEEE Access* **2021**, *9*, 125787–125804. https://doi.org/10.1109/ACCESS.2021.3111408.
30. Chen, C.; Chai, L.; Wang, Q. A wind power ultra-short-term ensemble forecast framework considering wind speed correction and scenario classification. *Energy* **2025**, *336*, 138276. https://doi.org/10.1016/j.energy.2025.138276.
31. Zhu, X.; Xu, Z.; Wang, Y.; Gao, X.; Hang, X.; Lu, H.; Liu, R.; Chen, Y.; Liu, H. Research on wind speed behavior prediction method based on multi-feature and multi-scale integrated learning. *Energy* **2023**, *263*, 125593. https://doi.org/10.1016/j.energy.2022.125593.
32. Dong, Y.; Zhou, B.; Zhang, H.; Yang, G.; Ma, S. A deep time-frequency augmented wind power forecasting model. *Renew. Energy* **2026**, *256*, 123550. https://doi.org/10.1016/j.renene.2025.123550.
33. Huang, Y.; Wang, Y.; Zou, R.; Aslam, L.; Hu, Q. Cycle-Aware Adaptive Mixture-of-Experts with High-Demand-Sensitive Asymmetric Loss for Effective Electrical Load Forecasting. *IEEE Trans. Power Syst.* **2026**, *41*, 4680–4695. https://doi.org/10.1109/TPWRS.2026.3703864.
34. Song, X.; Zhang, X.; Tian, W.; Zhu, Q. TFMSNet: A time series forecasting framework with time–frequency analysis and multi-scale processing. *Comput. Electr. Eng.* **2025**, *123*, 110260. https://doi.org/10.1016/j.compeleceng.2025.110260.

35. Liu, Z.; Cao, Y.; Xu, H.; Huang, Y.; He, Q.; Chen, X.; Tang, X.; Liu, X. Hidformer: Hierarchical dual-tower transformer using multi-scale mergence for long-term time series forecasting. *Expert Syst. Appl.* **2024**, *239*, 122412. https://doi.org/10.1016/j.eswa.2023.122412.
36. Song, Y.; Wan, R.; Li, L.; Wang, W.; Xing, H. A multi-scale temporal-frequency fusion network based on MLP for long-term time series forecasting. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 3943–3954. https://doi.org/10.1007/s13042-024-02491-y.
37. Xu, Y.; Yang, Y.; Che, J.; Jiang, Z.; Gan, Y. Multi-step wind speed forecasting based on multi-task learning using mixed granularity collaborative network. *Energy* **2026**, *345*, 140173. https://doi.org/10.1016/j.energy.2026.140173.
38. Cheng, R.; Shi, J. Collaborative wind speed forecasting based on multi-strategy MVMD and RestrictedCrossformer. *Energy* **2026**, *346*, 140163. https://doi.org/10.1016/j.energy.2026.140163.
39. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Advances in Neural Information Processing Systems; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 22419–22430. Available online: https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html (accessed on 5 August 2026).
40. Wang, Y.; Wu, H.; Dong, J.; Qin, G.; Zhang, H.; Liu, Y.; Qiu, Y.; Wang, J.; Long, M. TimeXer: Empowering transformers for time series forecasting with exogenous variables. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 469–498. https://doi.org/10.52202/079017-0015.
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 5998–6008. Available online: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed on 5 August 2026).
42. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325.
43. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency enhanced decomposed transformer for long-term time series forecasting. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 27268–27286. Available online: https://proceedings.mlr.press/v162/zhou22g.html (accessed on 5 August 2026).

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.