Environmetrics

SPECIAL ISSUE Exploring Statistical Strategies for Addressing Complex Environmental Challenges Involving Spatial-Temporal Data

RESEARCH ARTICLE

Spatial-Awared Deep Echo State Networks for High-Resolution Spatio-Temporal Wind Speed Nowcasting

Zipei Geng[1] | Stefano Castruccio[2] | Marc G. Genton[1]

[1]Statistics Program, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia | [2]Department of Applied and Computational Mathematics and Statistics, The University of Notre Dame, Notre Dame, Indiana, USA

Correspondence: Zipei Geng (zipei.geng@kaust.edu.sa)

Received: 2 June 2025 | **Revised:** 5 May 2026 | **Accepted:** 8 July 2026

Keywords: dimension reduction | echo state network | nearest neighbor | wind speed nowcasting

ABSTRACT

Accurate wind speed nowcasting is crucial for optimizing wind energy production and grid stability, especially for countries such as Saudi Arabia with ambitious renewable energy targets. This article introduces a novel deep learning framework combining Deep Echo State Networks (DESNs) with spatial information through k-nearest neighbors for high-resolution wind speed prediction. Specifically, we develop a Spatio-Temporal Attention Graph Autoencoder (STAGA) that effectively reduces spatial dimensionality while preserving critical spatio-temporal patterns, enabling efficient processing of large-scale meteorological data. Experiments using high-resolution simulated wind data from Saudi Arabia demonstrate that our model consistently outperforms traditional methods. In addition, we provide uncertainty quantification through conformal prediction and demonstrate its practical value by assessing wind power estimates. The proposed methodology offers significant improvements for operational wind forecasting systems, supporting the efficient integration of wind energy into power grids.

1 | Introduction

Wind energy plays a crucial role in the global transition toward renewable and sustainable energy sources (Blanco 2009). As countries seek to reduce their carbon footprint and mitigate the impacts of climate change, wind power has emerged as a promising alternative to fossil fuels (Kinley 2017). Saudi Arabia, despite its heavy dependence on oil reserves, has set an ambitious target of installing 16 GW of wind power capacity by 2030 as part of its "Vision 2030" plan (Nurunnabi 2017). To achieve this goal and ensure the optimal deployment of wind turbines, accurate forecasting of wind speeds is essential (Akpinar and Akpinar 2005; Spera 1994).

Wind speed nowcasting refers to the prediction of wind speed in the near future, typically within a few hours (Chkeir et al. 2023). Unlike traditional forecasting methods that focus on longer time horizons, nowcasting aims to provide high-resolution, short-term predictions that can help in the real-time operation and management of wind farms. Accurate wind speed nowcasting is crucial for optimizing power generation, reducing costs, and ensuring grid stability (Krishna et al. 2018; Cassola and Burlando 2012; Zhu and Genton 2012).

Modeling wind speed dynamics poses significant challenges due to the complex and nonlinear nature of atmospheric processes (Krishna et al. 2018; Cassola and Burlando 2012). Traditional statistical methods, such as autoregressive integrated moving average (ARIMA) models (Shumway and Stoffer 2017), have limitations in capturing the nonlinear, nonstationary spatio-temporal dependencies of wind speed. To address these limitations, machine learning techniques, particularly deep learning models, have gained prominence in wind speed forecasting.

Echo State Networks (ESNs, Jaeger (2007); McDermott and Wikle (2018); Bonas and Castruccio (2025)) have emerged as a promising approach for spatio-temporal forecasting tasks. ESNs belong to the family of reservoir computing (Nakajima and Fischer 2021) methods and are designed to efficiently model complex temporal dependencies. The key idea behind ESNs is to use a large, randomly initialized reservoir of neurons that remains fixed during training, while only the output weights are learned. This approach reduces computational burden and improves training stability compared to traditional recurrent neural networks (RNNs, Medsker and Jain (2001)). ESNs have been successfully applied to wind speed forecasting (McDermott and Wikle 2018; Huang et al. 2022; Bonas and Castruccio 2023, 2025), demonstrating their ability to capture nonlinear dynamics and provide accurate predictions. Other than ESNs, various deep learning architectures have been adapted for spatio-temporal modeling. Convolutional Long Short-Term Memory (ConvLSTM) networks (Shi et al. 2015) combine convolutional layers with LSTM (Hochreiter and Schmidhuber 1997) cells to capture both spatial and temporal dependencies at the same time. ConvLSTMs have shown promising results in wind speed forecasting, leveraging the ability of convolutional layers to extract local spatial features. Similarly, PredRNNs (Wang et al. 2017) integrate convolutional layers into RNNs to enhance the modeling of spatio-temporal patterns. In addition, attention mechanisms in spatio-temporal modeling aim to capture the most relevant spatial and temporal dependencies by assigning different weights to different spatial locations and time steps (Liu et al. 2019). Graph Convolutional Networks (GCNs, Zhang et al. (2019)) have also gained attention for their ability to model complex spatial relationships. GCNs operate on graph-structured data, where nodes represent spatial locations and edges represent the connections between them. Using the graph structure, GCNs can effectively capture spatial dependencies and propagate information across the graph. Although autoencoders (Tschannen et al. 2018) effectively compress high-dimensional data into compact representations, research specifically addressing spatial dimension reduction through convolutional autoencoders remains limited.

Despite advances in spatio-temporal deep learning models, developing accurate and computationally tractable models for high-resolution wind speed nowcasting remains a challenge, particularly in regions with complex geographies. In this work, we propose a novel ESN framework that models spatial dynamics using $k-$nearest neighbors in combination with deep ESN, which captures spatial and temporal dependence at the same time with the echo state property.

Our method builds on and improves previous work in nowcasting Saudi Arabia wind. Both Huang et al. (2022) and Wang et al. (2026) address spatial dimension reduction by selecting a subset of representative locations from the original observation grid—the former via domain-specific knot placement followed by covariance-based kriging reconstruction, the latter via energy-distance subsampling under the assumption of Gaussian process IID-ness in time with an SPDE-based spatial model. These approaches are effective and interpretable, but the spatial reduction is inherently tied to observed support locations and the reconstruction relies on a modeling stage that is decoupled from the forecasting objective. Our method differs from these in two key respects. First, rather than reducing the field to a pre-selected subset of spatial locations, we employ STAGA, a spatio-temporal graph autoencoder that learns a latent spatial representation directly from the full spatio-temporal field through graph convolution, temporal attention, and hierarchical pooling. The spatial compression is therefore entirely data-driven: no prior knot placement is required, and the encoder and decoder are trained end-to-end to optimize reconstruction fidelity, allowing the model to preserve nonlinear and spatially heterogeneous features that a fixed sparse knot set may not adequately summarize. Second, we use a deep ESN with nearest-neighbor inputs to model the spatio-temporal dynamics in the latent space, providing a nonlinear state-space model that captures spatial information directly in the forecasting stage rather than restricting spatial coupling to the reconstruction step. Together, these two aspects make our framework more flexible and model-agnostic, while retaining the computational advantages of the staged pipeline.

The article is organized into the following sections: an overview of the simulated dataset (Section 2), the proposed methodology (Section 3), the inference procedure (Section 4), the case study of wind speed nowcasting and uncertainty estimation in Saudi Arabia (Section 5), wind power predictions across 75 pre-determined wind farm locations (Section 6), and concluding remarks with discussion (Section 7).

2 | Wind Simulation Data

We use the dataset generated from a study conducted to perform hourly high-resolution simulations (Giani et al. 2020) using the Weather Research and Forecasting (WRF) model driven by the operational high-resolution European Center for Medium-Range Weather Forecast (ECMWF) model (Molteni et al. 1996). The simulations cover the period from 2013 to 2016, coinciding with the availability of wind speed data from meteorological masts for validation purposes.

The WRF model was simulated on a regular planar grid with fixed resolution using the Lambert Conformal Conic projection (Deetz 1918). Four different model setups were explored, and the one that most closely matched the observations was selected for this study. This setup utilized the Mellor-Yamada-Janjić planetary boundary layer scheme (Janić 2001) and had a spatial resolution of 6 km. The focus was on the hourly near-surface wind speed data.

For practical considerations, islands over the Red Sea (Tiran Island, Sanafir Island in the Red Sea Project Lagoon, and Farasan Island) were excluded from the analysis, resulting in a total of 53,333 locations (Huang et al. 2022). The 25%, 50%, and 75% percentile hourly near-surface wind speed from 2013 to 2016 are depicted in Figure 1, revealing generally higher values in the western regions due to the presence of mountain ranges. Although these areas are abundant in wind resources, they are often unsuitable for wind farm development due to the high installation costs associated with the rough terrain.

The northwest region of Saudi Arabia near the Gulf of Aqaba, which is the construction site of NEOM City (Farag 2019), exhibited favorable conditions for the harvest of wind energy based

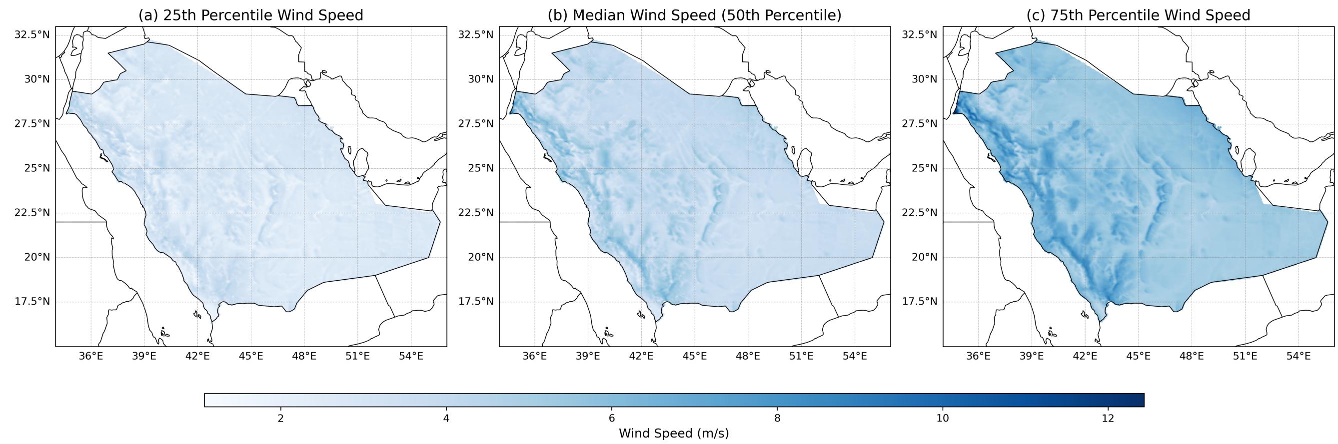

FIGURE 1 | 25%, 50%, and 75% percentile of daily simulation wind speed across time in Saudi Arabia from 2013 to 2016.

on the data analysis of Huang et al. (2022). Two locations with contrasting average wind speeds were selected to illustrate the diurnal temporal variability in terms of mean and standard deviation. It was observed that the wind speed annual variability in the Arabian Peninsula is relatively low, suggesting that the findings of this study could be extended beyond the simulation years (Giani et al. 2020).

The WRF model was validated against observations from 10 sites within the King Abdullah City for Atomic and Renewable Energy monitoring network using various assessment metrics. This validation process ensured the reliability and accuracy of the simulations for wind speed and wind power potential analysis in Saudi Arabia (Giani et al. 2020).

3 | Model

We provide a notation Table C1 to define the symbols and notation used throughout this article.

3.1 | Mean Structure

In this study, we investigate the periodic inter-daily and multi-daily patterns present in the hourly wind speed data across Saudi Arabia. These patterns can be attributed to daily land–ocean heat fluxes and mesoscale patterns of recurring winds. Specifically, to describe the dynamics of the square root of wind speed at time instance t and location $\mathbf{s}_i$, $Z_t(\mathbf{s}_i)$, where $i = 1, \ldots, N$, we use the following harmonic decomposition model:

$$\sqrt{Z_t(\mathbf{s}_i)} = \beta_0(\mathbf{s}_i) + \sum_{k=1}^{K} \left\{ \beta_{k,1}(\mathbf{s}_i) \cos\left(\frac{2\pi t}{T_k}\right) + \beta_{k,2}(\mathbf{s}_i) \sin\left(\frac{2\pi t}{T_k}\right) \right\} + \gamma(\mathbf{s}_i) Y_t(\mathbf{s}_i), \tag{3.1}$$

where K denotes the number of harmonics, $\beta_0(\mathbf{s}_i)$ represents the location-wise intercept, $\beta_{k,1}(\mathbf{s}_i)$ and $\beta_{k,2}(\mathbf{s}_i)$, $k = 1, \ldots, K$, are the location-wise coefficients for each harmonic with period T_k, and $\gamma(\mathbf{s}_i)$ is the location-wise scaling parameter such that the residual $Y_t(\mathbf{s}_i)$ has unit variance at each location. Preliminary studies indicate that there are $K = 5$ significant harmonics at periods of 1 year, half a year, 1 day, 12 h, and 8 h. The square root transformation is applied to normalize the wind speed data, which are generally right-skewed due to occasional wind gusts. In the upcoming sections, we focus on capturing the spatio-temporal patterns present in the wind speed residuals, $Y_t(\mathbf{s}_i)$.

3.2 | Wind Residuals

Our main task is to accurately capture the spatial and temporal dependence within the wind residuals given the mean structure introduced in Equation (3.1). The hourly wind residual $Y_t(\mathbf{s}_i)$, $i = 1, \ldots, N$, is expected to exhibit highly nonlinear behavior that traditional time series approaches, such as ARIMA or more general linear space–time models, cannot capture effectively. Furthermore, due to the nature of our spatio-temporal dataset, in which the spatial locations are irregularly spaced, we propose a new model which utilizes the spatial information of k-nearest neighbors. This idea coincides with the graph convolutional network, where its inputs are the pair of spatio-temporal data vector $\mathbf{x}$ and its adjacency matrix $\mathbf{A}$, where $\mathbf{A} := (\delta_{ij})$, with $\delta_{ij} = 1$, if i is one of the k-nearest neighbors of j, or $\delta_{ij} = 0$ otherwise. Graph Neural Networks (GNNs) (Zhang et al. 2019) represent a standard approach for handling irregularly distributed data, which exhibits cubic computational complexity as the number of spatial locations increases. Consequently, spatial dimension reduction becomes necessary. To address this, we propose a framework that addresses the reduction of spatial dimensions, nonlinear dynamical processes, and spatial correlation issues.

In the subsequent sections, we address the nowcasting of wind residuals $\mathbf{y}_t = \left(Y_t(\mathbf{s}_1), \ldots, Y_t(\mathbf{s}_N)\right)^\top$ utilizing historical observations within a specified temporal window that excludes information beyond a predetermined relevance threshold. Specifically, we perform the following steps:

1. We initially compress the spatial dimension of $\mathbf{y}_t$ through a graph autoencoder, a neural network architecture that reduces high-dimensional input to lower-dimensional representations while retaining key spatio-temporal features. The autoencoder employs an encoder to project input data into a compact latent space and a decoder to recover the original data from this representation, thereby capturing

essential data structure while minimizing computational overhead.

2. The compressed spatio-temporal data are subsequently processed using a deep ESN (DESN) that integrates k-nearest neighbor spatial information as DESN inputs, which strengthens the model's capacity to learn spatial dependencies and enhances prediction accuracy by enriching the spatio-temporal information available to the regularized regression predictors.
3. The trained autoencoder is then employed to reconstruct the complete spatial field of wind residuals $\mathbf{y}_t$ from the compressed representation. This reconstruction process enables high-resolution wind residual predictions across all spatial locations for subsequent analysis and decision-making applications.

3.3 | Deep Echo State Networks

Here we review the DESN architecture. A DESN is formulated as follows (McDermott and Wikle 2018):

$$\text{output: } \mathbf{y}_t = \mathbf{B}_D g(\mathbf{h}_{t,D}) + \sum_{d=1}^{D-1} \mathbf{B}_d g(\tilde{\mathbf{h}}_{t,d}) + \epsilon_t, \tag{3.2}$$

$$\text{hidden state } d: \mathbf{h}_{t,d} = (1-\phi)\mathbf{h}_{t-1,d} + \phi f\left(\frac{\delta_d}{|\lambda_{\mathbf{W}_d}|}\mathbf{W}_d \mathbf{h}_{t-1,d} + \mathbf{V}_d \tilde{\mathbf{h}}_{t,d-1}\right), \text{ for } d > 1, \tag{3.3}$$

$$\text{reduction } d-1: \tilde{\mathbf{h}}_{t,d-1} = F(\mathbf{h}_{t,d-1}), \text{ for } d > 1, \tag{3.4}$$

$$\text{input: } \mathbf{h}_{t,1} = f\left(\frac{\delta_1}{|\lambda_{\mathbf{W}_1}|}\mathbf{W}_1 \mathbf{h}_{t-1,1} + \mathbf{V}_1 \mathbf{x}_t\right), \tag{3.5}$$

where $\mathbf{x}_t = (1, \mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-m})^\top$ is an $(mN+1)-$dimensional vector and D is the number of layers. Here, $\mathbf{B}_d, d = 1, \ldots, D$ are unknown coefficient matrices that will be estimated using ridge regression with penalty λ_r. The vectors $\mathbf{h}_{t,d}, d = 1, \ldots, D$ are hidden states at time t that are updated using the combination of hidden states of the same layer at time $t-1$ and dimension-reduced hidden states $\tilde{\mathbf{h}}_{t,d-1}$ of the previous layer at time t, where the dimension reduction function $F(\cdot)$ is taken as principal component analysis (PCA) in this article. A normalization function $g(\cdot)$ is used to ensure consistent variability ranges across layers. Here ϕ is the leakage rate which determines how many percent of historical hidden states one wishes to include. Furthermore, to imitate the process of feed-forward neural network, we define $f(\cdot)$ as element-wise rectified linear unit (ReLU) function. δ_d are tuning parameters and $\lambda_{\mathbf{W}_d}$ is the largest eigenvalue with respect to randomly generated matrices $\mathbf{W}_d$ because of the echo state property, where the matrix distribution is given by

$$(\mathbf{W}_d)_{ij} = \gamma_{ij}^{\mathbf{W}_d} U(-a_{\mathbf{W}_d}, a_{\mathbf{W}_d}), \gamma_{ij}^{\mathbf{W}_d} \sim \text{Bern}(\pi_{\mathbf{W}_d}).$$

Also, the matrices $\mathbf{V}_d, d = 1, \ldots, D$ are randomly generated according to the following rule:

$$(\mathbf{V}_d)_{ij} = \gamma_{ij}^{\mathbf{V}_d} U(-a_{\mathbf{V}_d}, a_{\mathbf{V}_d}), \gamma_{ij}^{\mathbf{V}_d} \sim \text{Bern}(\pi_{\mathbf{V}_d}).$$

Here $a_{\mathbf{V}_d}$ and $a_{\mathbf{W}_d}$ are the magnitudes of the matrices that are about to be tuned. Finally, $g(\cdot)$ is an element-wise hyperbolic tangent function to unify the scale of covariates, which will be the input of the Ridge regression (3.2).

The dynamical reservoir is similar to an RNN (Medsker and Jain 2001) architecture with randomly sampled weights rather than using backpropagation Werbos (1990) to optimize.

3.4 | k-Nearest Neighbor Implementation

We employ a spatial convolution filter as the input layer of the DESN, functioning as an auxiliary reservoir that establishes connections between neighboring spatial locations and the hidden states. We define the $k-$connectivity of two spatial locations when one spatial location belongs to the $k-$nearest neighboring set of the other spatial location. Under this connectivity, we add spatial locations directly to the input $\mathbf{x}_t$ as defined in the previous section.

We define k-nearest neighbors through an undirected graph $\mathbf{g}$ with vertices $V(\mathbf{g})$ and edges $E(\mathbf{g})$. Vertices are the spatial locations where we observed the wind speed data: $E(\mathbf{g}) := \{(u, v) : u, v \in V(\mathbf{g})\}$. The number of vertices of $\mathbf{g}$ is denoted by $|V(\mathbf{g})|$. Given any $v \in V(\mathbf{g})$, define the ordered (from nearest to farthest) neighbor list of v (excluding v) as $\mathcal{ON}(v) := \{u_1^v, u_2^v, \ldots, u_{|V(\mathbf{g})|-1}^v\}$, where $\|u_i^v - v\|_2 \le \|u_j^v - v\|_2$, if $i < j$. The $k-$connectivity is thus defined as follows:

$$(u, v) \in E(\mathbf{g}) \quad \text{and} \quad u \in \mathcal{ON}(v)[1:k],$$

where $[1:k]$ the first 1 to k elements of an ordered list. Given the above definitions, we define the neighborhood list of v as $\mathcal{N}(v) := \{u \in V(\mathbf{g}) : (u, v) \in E(\mathbf{g})\}$.

Then the nearest neighbor filter model can be defined as follows. Given the input formula (3.5), we have the $k-$connectivity adapted version

$$\mathbf{h}_{t,1} = f\left(\frac{\delta_1}{|\max(\lambda_{\mathbf{W}_1}, \lambda_{\mathbf{W}_1'})|}\widetilde{\mathbf{W}}\tilde{\mathbf{h}}_{t-1,1} + \begin{bmatrix}\mathbf{V}_1\\ \mathbf{M}\end{bmatrix}\left[\mathbf{x}_t, \mathbf{x}_t(\mathcal{N})\right]\right), \tag{3.6}$$

where

$$\widetilde{\mathbf{W}} = \begin{pmatrix}\mathbf{W}_1 & \mathbf{O}_{n_{h,1}\times n_f}\\ \mathbf{O}_{n_f\times n_{h,1}} & \mathbf{W}_1'\end{pmatrix} \in \mathbb{R}^{(n_{h,1}+n_f)\times(n_{h,1}+n_f)}, \quad \tilde{\mathbf{h}}_{t-1,1} \in \mathbb{R}^{(n_{h,1}+n_f)\times 1},$$

$$\mathbf{V}_1 \in \mathbb{R}^{n_{h,1}\times(mN+1)}, \quad \mathbf{x}_t \in \mathbb{R}^{(mN+1)\times 1}, \quad \mathbf{x}_t(\mathcal{N}) \in \mathbb{R}^{(kmN+1)\times 1},$$

and n_f is the number of nearest neighbor filters to add to the original DESN model. $\mathbf{x}_t(\mathcal{N})$ is defined as

$$\left[\mathbf{y}_{t-1}^\top(\mathcal{N}), \mathbf{y}_{t-2}^\top(\mathcal{N}), \ldots, \mathbf{y}_{t-m}^\top(\mathcal{N})\right],$$

where $\mathbf{y}_{t-i}^\top(\mathcal{N}) = (y_{t-i}^\top(\mathcal{N}(s_1)), \ldots, y_{t-i}^\top(\mathcal{N}(s_N))^\top)$ and $y_{t-i}^\top(\mathcal{N}(s_j)) = (y_{t-i}(\mathcal{ON}(s_j)[1]), \ldots, y_{t-i}(\mathcal{ON}(s_j)[k])$.

The idea behind this model is to encode spatial neighboring information from the input to the reservoir. The neighboring

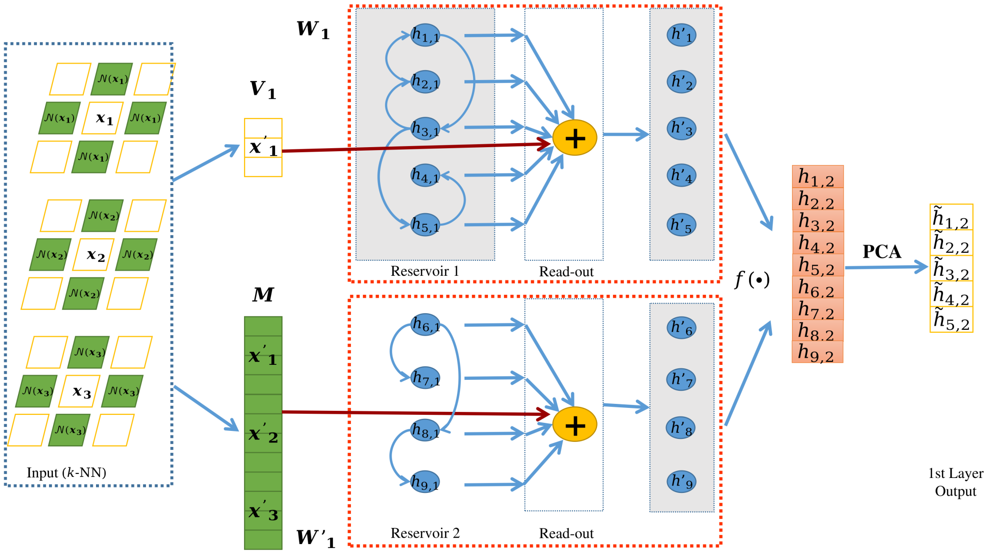

FIGURE 2 | Schematic description of the 1st layer of SA-DESN model. We fix time at t' and use $h_{i,1}$ to denote hidden states at dimension (of the vector) i and layer 1. The number of hidden states for reservoir 1 is $n_h = 5$, the number of hidden states for reservoir 2 is $n_f = 4$, with the number of nearest neighbors $k = 4$. The blue-colored output of read-out $h'_{i,1}$ is then fed into activation function f. The orange-colored activated output undergoes dimensionality reduction through PCA to produce $\tilde{h}_{i,1}$, which serves as the input for the subsequent layer. This iterative process continues through all layers up to the D-th layer. The output of the D-th layer is then used to predict the wind residuals at time t'.

sets can possess different features that could help to potentially increase the predictability of our model. We choose to encode these neighboring sets into the RNN input layer. The number of filters is defined as n_f. To fulfill the echo state property (Yildiz et al. 2012), we require the largest eigenvalue of the matrix $\widetilde{\mathbf{W}}$ to be less than 1, which is achieved by scaling the matrix $\widetilde{\mathbf{W}}$ by $\delta_1 / |\max(\lambda_{\mathbf{W}_1}, \lambda_{\mathbf{W}'_1})|$. To clarify, since there are two separate reservoirs that must each satisfy the echo state property, it is adequate to use the maximum of the largest eigenvalues from both transfer matrices.

Finally, the matrix $\mathbf{M}$ is the transfer matrix between spatial information and the neighboring set, which will be of dimension $n_f \times kmN$. This transfer matrix is allowed to enclose sparsity to avoid overfitting. We have

$$\mathbf{M}_{ij} | \gamma_{ij}^{\mathbf{M}} \sim \gamma_{ij}^{\mathbf{M}} U(-a_{\mathbf{M}}, a_{\mathbf{M}}), \quad \gamma_{ij}^{\mathbf{M}} \sim \text{Bern}(\pi_{\mathbf{M}}).$$

We combine the idea of the k−nearest neighbor filter into a DESN framework with D layers. The complete model is called Spatial-Awared Deep Echo State Network (SA-DESN); see Figure 2 for a schematic description.

3.5 | Spatio-Temporal Attention Graph Autoencoder (STAGA) for Wind Speed Modeling

We propose a spatio-temporal attention graph autoencoder (STAGA) for wind speed data compression and reconstruction based on the idea of Graph U-Nets (Gao and Ji 2019) and Graph Attention Network (Veličković et al. 2018). The model leverages graph neural networks to reduce spatial complexity while preserving important temporal patterns.

Let $\mathbf{X} \in \mathbb{R}^{N \times T}$ denote the spatio-temporal wind speed data, where $x_t(\mathbf{s}_i) \in \mathbb{R}$ represents wind speed at location $\mathbf{s}_i$ at time t. We aim to learn a spatial compression function $\mathcal{E} : \mathbb{R}^{N \times T} \to \mathbb{R}^{n \times T}$ and a spatial reconstruction function $\mathcal{D} : \mathbb{R}^{n \times T} \to \mathbb{R}^{N \times T}$ such that

$$\hat{\mathbf{X}} = \mathcal{D}(\mathcal{E}(\mathbf{X})) \approx \mathbf{X}$$

where $n \ll N$ represents the compressed spatial dimension while preserving the full temporal dimension T.

Define an undirected graph $\mathcal{G} = (V, E, \mathbf{A})$ where

- $V = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\}$ represents the spatial locations
- $E \subseteq V \times V$ represents edges based on spatial proximity
- $\mathbf{A} \in \{0, 1\}^{N \times N}$ is the adjacency matrix constructed using k-nearest neighbors with Euclidean distance.

3.5.1 | Skeleton of the STAGA

STAGA processes each time step independently through the spatial compression network while using temporal attention to capture dependencies. For each time step t: $\mathbf{z}_t = \mathcal{E}_{\text{spatial}}(\check{\mathbf{x}}_t, \mathcal{G}, \mathbf{X}_{t-W:t}) \in \mathbb{R}^n$ where $\check{\mathbf{x}}_t = [x_t(\mathbf{s}_1), \dots, x_t(\mathbf{s}_N)]^\top$ and $\mathbf{X}_{t-W:t}$ is the temporal context window. For each spatial location $\mathbf{s}_i$ and time t, we extract temporal features using LSTM: $\mathbf{f}_{i,t}^{\text{temp}} = LSTM(\check{\mathbf{x}}_t) \in \mathbb{R}^{d_f}$. Then, we design a temporal window for attention: $\mathbf{R}_{i,t} = [\mathbf{f}_{i,t-W+1}^{\text{temp}}, \dots, \mathbf{f}_{i,t}^{\text{temp}}] \in \mathbb{R}^{W \times d_f}$.

3.5.2 | Encoder Architecture

At each time step t, the scalar wind speed residual values are projected $\mathbf{f}_{i,t}^{(0)} = \mathbf{W}_{\text{proj}} x_t(\mathbf{s}_i) + \mathbf{b}_{\text{proj}} \in \mathbb{R}^{d_f}$.

The encoder operates on the spatial graph at each time step t, whose implementation is summarized in Algorithm 1. The

ALGORITHM 1 | Spatio-temporal attention encoder.

Input: $\mathbf{F}_t^{(0)} \in \mathbb{R}^{N\times d_f}$, $\mathbf{A}^{(0)} \in \mathbb{R}^{N\times N}$, $\mathbf{R}_t \in \mathbb{R}^{N\times W\times d_f}$
Initialize: $\mathcal{F}_t = \emptyset$, $\mathcal{A} = \emptyset$, $\mathcal{P} = \emptyset$
for $\ell = 1$ to L **do**
 // Graph convolution on spatial dimension
 $\mathbf{F}_t^{(\ell)} = \mathrm{GCN}^{(\ell)}(\mathbf{F}_t^{(\ell-1)}, \mathbf{A}^{(\ell-1)})$
 // Temporal attention enhancement
 $\mathbf{F}_t^{(\ell)} = \mathbf{F}_t^{(\ell)} + \mathrm{TemporalAttention}^{(\ell)}(\mathbf{F}_t^{(\ell)}, \mathbf{R}_t^{(\ell-1)})$
 Store: $\mathcal{F}_t[\ell] = \mathbf{F}_t^{(\ell)}$, $\mathcal{A}[\ell] = \mathbf{A}^{(\ell-1)}$
 if $\ell < L$ **then**
  //Spatial pooling
  $\mathbf{F}_t^{(\ell)}, \mathbf{A}^{(\ell)}, \mathbf{p}^{(\ell)} = \mathrm{TopKPool}(\mathbf{F}_t^{(\ell)}, \mathbf{A}^{(\ell-1)}, r)$
  Store: $\mathcal{P}[\ell] = \mathbf{p}^{(\ell)}$
  //Update temporal window for reduced spatial nodes
  $\mathbf{R}_t^{(\ell)} = \mathbf{R}_t^{(\ell-1)}[\mathbf{p}^{(\ell)}, :, :]$
 end if
end for
Return: $\mathbf{F}_t^{(L)} \in \mathbb{R}^{n\times d_f}$ where $n = \lceil r^L \cdot N \rceil$

encoder consists of L layers, where each layer performs graph convolution followed by temporal attention enhancement.

The graph convolutional network (GCN) at time t is defined as

$$\mathbf{F}_t^{(\ell)} = ReLU(\tilde{\mathbf{A}}^{(\ell-1)}\mathbf{F}_t^{(\ell-1)}\mathbf{W}_{\mathrm{GCN}}^{(\ell)}),$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-1/2}$, $\mathbf{D}$ is the degree matrix, and $\mathbf{W}_{\mathrm{GCN}}^{(\ell)}$ are learnable parameters. Then, we define the TopKPool function, which will be used as a graph pooling technique. The pooling operation selects the most informative spatial locations based on the score, defined as

$$\mathrm{score}_{i,t} = \frac{\mathbf{f}_{i,t}^{(\ell)}\mathbf{p}^{(\ell)}}{\|\mathbf{f}_{i,t}^{(\ell)}\|\|\mathbf{p}^{(\ell)}\|}$$

At layer ℓ, the spatial node indices are then given by the largest scores

$$\mathrm{idx}_t^{(\ell)} = \mathrm{TopK}(\mathrm{score}_t[:], \lceil r \cdot N^{(\ell-1)}\rceil),$$

where $\mathbf{p}^{(\ell)}$ is a learnable projection vector, $r \in (0,1)$ is the pooling ratio, and $N^{(\ell-1)}$ is the number of nodes at layer $\ell - 1$.

At each encoder layer ℓ and time t, we then apply temporal attention to capture relevant historical patterns. Given the recent temporal window $\mathbf{R}_t^{(\ell)}$, for each spatial node $\mathbf{s}_i$, we have that

$$\mathbf{Q}_{i,t} = \mathbf{W}_Q^{(\ell)}\mathbf{f}_{i,t}^{(\ell)} \in \mathbb{R}^{d_f\times 1}$$
$$\mathbf{K}_{i,t} = \mathbf{R}_{i,t}^{(\ell)}\mathbf{W}_K^{(\ell)} \in \mathbb{R}^{W\times d_f}$$
$$\mathbf{V}_{i,t} = \mathbf{R}_{i,t}^{(\ell)}\mathbf{W}_V^{(\ell)} \in \mathbb{R}^{W\times d_f}$$

$$\alpha_{i,t,w} = \frac{\exp\left(\frac{\mathbf{Q}_{i,t}^\top \mathbf{K}_{i,t,w}}{\sqrt{d_h}} + b_w\right)}{\sum_{j=1}^{W}\exp\left(\frac{\mathbf{Q}_{i,t}^\top \mathbf{K}_{i,t,j}}{\sqrt{d_f}} + b_j\right)}$$

where $b_w = \frac{2(w-1)}{W-1}$ encourages attention to recent observations. The attended feature is then given by $\mathbf{f}_{i,t}^{\mathrm{att}} = \sum_{w=1}^{W}\alpha_{i,t,w}\mathbf{V}_{i,t,w}$.

After L levels of spatial compression, we obtain the bottleneck $\mathbf{Z}^* = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T] \in \mathbb{R}^{n\times T}$, where each $\mathbf{z}_t = \mathbf{F}_t^{(L)} \in \mathbb{R}^{n\times d_f}$ is flattened to $\mathbb{R}^n$ using fully connected layer with tanh activation.

3.5.3 | Decoder Architecture

For each time step t, we reconstruct the spatial resolution as $\mathbf{F}_{t,\mathrm{unpool}}^{(\ell)}[\mathrm{idx}_t^{(\ell)}] = \mathbf{F}_t^{(\ell+1)}$. The unpooled features are concatenated with encoder features: $\mathbf{F}_{t,\mathrm{concat}}^{(\ell)} = [\mathbf{F}_{t,\mathrm{unpool}}^{(\ell)}, \mathbf{F}_{t,\mathrm{enc}}^{(\ell)}] \in \mathbb{R}^{N^{(\ell)}\times 2d_f}$ and passed through a graph convolutional network (GCN) to reconstruct the spatial features at each layer: $\mathbf{F}_{t,\mathrm{dec}}^{(\ell)} = GCN_{\mathrm{dec}}^{(\ell)}(\mathbf{F}_{t,\mathrm{concat}}^{(\ell)}, \mathbf{A}^{(\ell)})$. The final spatial reconstruction at each time t is given by $\hat{x}_t(\mathbf{s}_i) = \mathbf{W}_{\mathrm{out}}\mathbf{f}_{i,t,\mathrm{dec}}^{(0)} + b_{\mathrm{out}}$,where $\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{1\times d_f}$.

The reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^{S\times T}$ is formed by concatenating the results across all time steps, with reconstruction error given by

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N\cdot T}\sum_{i=1}^{N}\sum_{t=1}^{T}(x_t(\mathbf{s}_i) - \hat{x}_t(\mathbf{s}_i))^2.$$

The compact representation $\mathbf{Z}^* \in \mathbb{R}^{N\times T}$ can be directly fed into SA-DESN for future prediction

$$\mathbf{Z}_{T+1}^*, \mathbf{Z}_{T+2}^*, \ldots = \mathrm{SA\text{-}DESN}(\mathbf{Z}^*) \in \mathbb{R}^{n\times h}$$

and the full-dimensional wind speed residuals are reconstructed by the trained decoder.

3.5.4 | Relative Merits of STAGA

STAGA relies on a spatial compression approach which is substantially different from the knot-based dimension reduction used in Huang et al. (2022) and Wang et al. (2026). In those approaches, the field is compressed to a discrete set of pre-selected locations (chosen either by domain-expert design or by energy-distance subsampling) and the full field is subsequently recovered via a separately fitted spatial model (kriging or SPDE). In STAGA the nodes are selected based on learned attention scores that reflect the observed spatio-temporal structure of the data rather than any prior on spatial representativeness. Crucially, the decoder is trained jointly with the encoder under the same reconstruction objective, so the latent representation is optimized end-to-end for reconstruction fidelity rather than through a decoupled interpolation stage. This design allows the encoder to retain information about nonlinear and spatially heterogeneous features of the wind field that may be poorly approximated by a sparse fixed set of knots.

However, as such, the manifold learned by STAGA does not correspond to physical locations, nor does it have any interpretable spatial features. Instead, the abstract learned representation is optimized for capturing the dominant spatio-temporal structure in the data and does not carry any physical interpretation.

3.6 | Pipeline

After introducing the basic architecture of STAGA, we introduce the pipeline of our residual wind speed nowcasting. The pipeline

ALGORITHM 2 | STAGA with SA-DESN (SA-DESN-STAGA) pipeline.

Require:
Wind residual data matrix $\mathbf{X} \in \mathbb{R}^{N\times T}$, spatial coordinates matrix $\mathbf{P} \in \mathbb{R}^{N\times 2}$ (lon, lat)
Number of nearestneighbors k, target reduced dimension n, forecast horizon h
Ensure:
Forecasted wind residuals at all N locations
Phase 1: Dimension Reduction with STAGA
Construct k-nearest neighbor graph adjacency matrix $\mathbf{A}$ from $\mathbf{P}$
Initialize STAGA parameters θ_{STAGA}
$\mathcal{L} \leftarrow \infty$ ▷ Initialize reconstruction loss
while not converged **do**
  $\mathbf{Z}^* \leftarrow \text{Encoder}_{\theta_{STAGA}}(\mathbf{X}, \mathbf{A})$ ▷ $\mathbf{Z}^* \in \mathbb{R}^{N\times T}$
  $\hat{\mathbf{X}} \leftarrow \text{Decoder}_{\theta_{STAGA}}(\mathbf{Z}^*, \mathbf{A})$
  $\mathcal{L} \leftarrow \frac{1}{N\cdot T}\sum_{i=1}^{N}\sum_{t=1}^{T}(x_t(\mathbf{s}_i) - \hat{x}_t(\mathbf{s}_i))^2$
  Update θ_{STAGA}using backpropagation to minimize $\mathcal{L}$
end while
Extractreduced representations $\mathbf{Z}^* \in \mathbb{R}^{n\times T}$
Store encoder and decoder parameters θ^*_{STAGA}
Phase 2: Spatio-Temporal Forecasting with SA-DESN
Split $\mathbf{Z}^*$ into training and testing sets
Train SA-DESNand tune hyperparameters on reduced representations $\mathbf{Z}^*$using the split training set (train → train + validation) and timeseries cross-validation technique (introduce later)
for each time step t in test set **do**
  $\hat{\mathbf{Z}}_{t+1:t+h} \leftarrow \text{SA-DESN}_{\theta_{SA-DESN}}(\mathbf{Z}^*_{t-m:t})$ ▷ Forecast h steps ahead using window m
end for
Phase 3: Reconstruction of Full-Dimensional Forecasts
for each forecasted time step $t+i, i \in \{1, \ldots, h\}$ **do**
  $\hat{\mathbf{X}}_{t+i} \leftarrow \text{Decoder}_{\theta^*_{STAGA}}(\hat{\mathbf{Z}}_{t+i}, \mathbf{A})$ ▷ Map back to full spatial dimension
end for
return Forecasted full-dimensional wind residuals $\hat{\mathbf{X}}_{t+1:t+h}$

is illustrated by the following Algorithm 2. The main idea is to first reduce the spatial dimension of the original data using STAGA, and then use SA-DESN to predict the wind speed residuals. Finally, we reconstruct the full-dimensional wind speed residuals using the learned decoder from STAGA. The pipeline is shown in Figure 3.

4 | Inference

4.1 | Estimating the Mean Structure

From mean structure model (3.1), the parameters $\beta_0(\mathbf{s}_i)$, $\beta_{k,1}(\mathbf{s}_i)$, and $\beta_{k,2}(\mathbf{s}_i)$, $i = 1, \ldots, N$, $k = 1, \ldots, K$, are estimated by linear regression independently across sites using the data from 2013 to 2014 (training data), a task that can be performed in parallel across locations. The scaling parameter $\gamma(\mathbf{s}_i)$, $i = 1, \ldots, N$, is also estimated site by site so that the resulting $Y_t(\mathbf{s}_i)$ is a zero-mean spatio-temporal residual process.

4.2 | Estimating the Temporal Structure

To elaborate and provide additional details regarding Equation (3.2), we present the ridge regression approach with regularization parameter λ_r. First, we define the response $\mathbf{Y}$, covariates $\mathbf{H}$, and error term $\mathbf{E}$ as follows:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_T^\top \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1^\top \\ \vdots \\ \mathbf{h}_T^\top \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \boldsymbol{\varepsilon}_1^\top \\ \vdots \\ \boldsymbol{\varepsilon}_T^\top \end{bmatrix},$$

where

$$\mathbf{h}_t^\top = (\mathbf{h}_{t,D}, g(\tilde{\mathbf{h}}_{t,D-1}), \ldots, g(\tilde{\mathbf{h}}_{t,1}))^\top, \quad \text{for} \quad t = 1, \ldots, T.$$

The regression formula can be expressed as

$$\mathbf{Y} = \mathbf{H}\mathbf{B} + \mathbf{E},$$

where $\mathbf{B} = (\mathbf{B}_D, \mathbf{B}_{D-1}, \ldots, \mathbf{B}_1)^\top$. The ridge estimator is then given by $\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \left(\|\mathbf{Y} - \mathbf{H}\mathbf{B}\|_F^2 + \lambda_r\|\mathbf{B}\|_F^2\right)$, where $\|\cdot\|_F$ is the Frobenius norm. We will use this formula recursively to fine-tune the hyperparameters using specialized time series cross-validation in Section 4.3, and the final model will be selected based on the performance of the test split.

4.3 | Time Series Cross-Validation

The hyperparameter tuning process for the time series model involves a rigorous validation methodology respecting the temporal nature of the data. We split the dataset into training and test data, with training further split into train and validation sets. Let $\mathbf{y}_t$ represent our multivariate time series observations

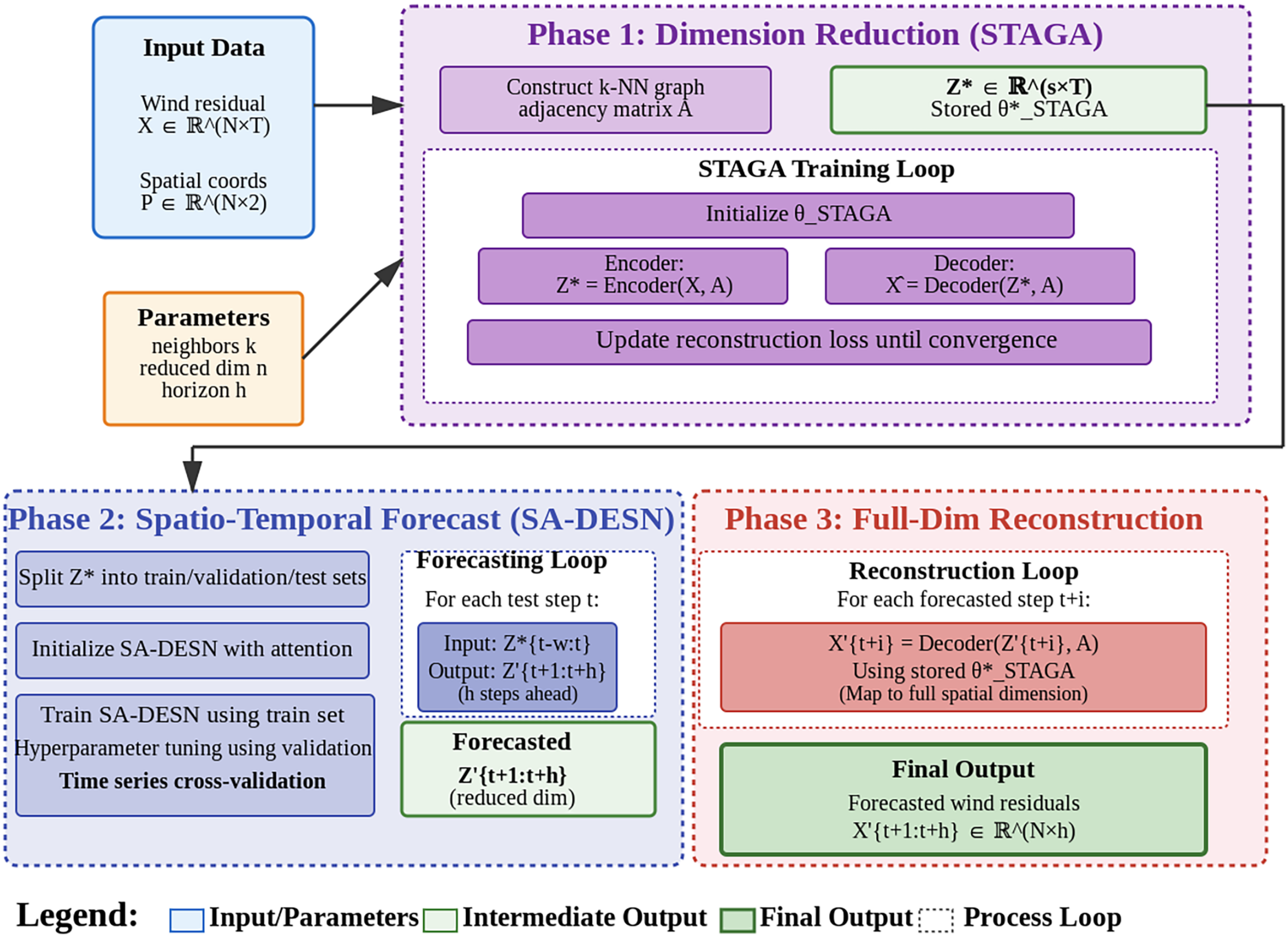

FIGURE 3 | SA-DESN-STAGA pipeline for wind speed nowcasting. Phase 1: Dimension reduction with STAGA; Phase 2: Forecasting with SA-DESN; Phase 3: Reconstruction of full-dimensional forecasts.

with temporal length T. Several parameters require estimation via cross-validation: the reservoir state dimension $n_{h,d}$ at layer d, the number of nearest neighbor filters n_f, the number of nearest neighbors k, the number of lags m in the input vector $\mathbf{x}_t$, the leaking rate ϕ and the scaling parameter δ_d in Equations (3.3) and (3.5), the magnitude and sparsity parameters $a_{\mathbf{M}}, \pi_{\mathbf{M}}, a_{\mathbf{W}_d}$, $\pi_{\mathbf{W}_d}$, $a_{\mathbf{W}'_1}$, $\pi_{\mathbf{W}'_1}$ $a_{\mathbf{V}_d}$, and $\pi_{\mathbf{V}_d}$. We employ an expanding window validation scheme. We partition this sequence into C consecutive validation folds of equal size v, where the i-th validation fold is defined as $\mathcal{V}_i = [t_{i-1}+1, t_i]$ with $t_i = t_{i-1} + v$ and $t_0 = T_0$, where T_0 represents the initial training period. For each fold $i \in \{1, \ldots, C\}$, we define an expanding training window $\mathcal{TW}_i = [t_0, t_i]$ that includes all observations prior to the validation fold. This approach ensures that as i increases, the training window expands to incorporate more historical data while maintaining the temporal integrity of the prediction task. For a given hyperparameter configuration $\boldsymbol{\theta}$, we train a model on each $\mathcal{TW}_i$ and evaluate its performance on the corresponding $\mathcal{V}_i$, yielding the validation score:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{|\mathcal{V}_i|}\sum_{t\in\mathcal{V}_i}\mathrm{MSE}_{1,2,3}(\mathbf{y}_t, \hat{\mathbf{y}}_{t|\mathcal{TW}_i}(\boldsymbol{\theta})), \tag{4.1}$$

where $\hat{\mathbf{y}}_{t|\mathcal{TW}_i}(\boldsymbol{\theta})$ represents the prediction for time t using the model trained on $\mathcal{TW}_i$ with hyperparameters $\boldsymbol{\theta}$, and $\mathrm{MSE}_{1,2,3}$ is the arithmetic mean of MSE across 1–3 h lead predictions. The optimal hyperparameter configuration is selected as $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$. After the tuning process is over, we use $\boldsymbol{\theta}^*$ with the entire training data set to obtain the final $\hat{\mathbf{B}}^*_d$, $d = 1, \ldots, D$ for prediction.

All experiments were conducted across three dedicated computing environments, allocated according to the computational profile of each pipeline stage.

STAGA training (Phase 1) was performed on a server equipped with two NVIDIA Ampere A100 GPUs (80 GB HBM2e memory each). The full spatial graph of $N = 53,333$ nodes is processed in temporal mini-batches; at each training step, the graph adjacency matrix, node feature tensors, and intermediate GNN activations collectively occupy approximately 40–60 GB of GPU memory per device, making the 80 GB capacity necessary.

SA-DESN hyperparameter tuning (Phase 2) was distributed across four NVIDIA Volta V100 GPUs (32 GB GDDR6 memory each) hosted on two separate servers (two GPUs per server). The tuning procedure follows the expanding-window time series cross-validation, evaluating a candidate hyperparameter grid over C validation folds. Because each combination of hyperparameters and each of the $C = 100$ ensemble members involves an independent forward pass through the reservoir and a closed-form ridge regression solve, the workload is embarrassingly parallel across ensemble members and hyperparameter candidates. Jobs were distributed across the four V100 devices using a task-queue parallelism strategy, with each GPU processing one ensemble–fold pair at a time.

CPU-intensive tasks, including inference for Kriging-based baseline methods, conformal prediction calibration, and all post-processing and visualization, were performed on a workstation equipped with an AMD Ryzen Threadripper 3995WX processor (64 cores, 128 threads, base clock 2.7 GHz). Kriging interpolations for the ESN-Kriging, AESN-Kriging, and SA-DESN-Kriging baselines, which require covariance matrix operations over the 3173 selected knots, were handled on this machine.

5 | Results

We now demonstrate the significant advantages of SA-DESN for wind speed forecasting across different prediction horizons. Tables 1 and 2 provide evidence of the better performance achieved by our proposed models compared to traditional approaches. The global setting for ESN prediction is to use 100 ensembles means. The training period is from 2013 to 2014, the validation period 2015, and the testing period is 2016. The training data are split into train and validation sets, with the validation set used for hyperparameter tuning with our proposed validation method. The final model is trained on the entire training set, and the performance is evaluated on the test set.

We compare our proposed models (boldfaced) with the following:

- Persistence: Persistence model ($\mathbf{y}_{t+1} = \cdots = \mathbf{y}_{t+h} = \mathbf{y}_t$)
- ESN-Kriging: Echo State Network with Kriging interpolation (Huang et al. 2022) with 3173 knots selected for spatial Kriging
- AESN-Kriging: Areal Echo State Network (Wang et al. 2025) (details can be found in Supplementary D) with Kriging (based on 3173 selected knots)
- SA-DESN-Kriging: Spatial-Awared DESN with Kriging interpolation (based on 3173 selected knots)
- SA-DESN-STAGA: Spatial-Awared DESN with STAGA (based on $n = 2500$ reduced spatial locations, we use the same number of n while using STAGA to reduce spatial dimension). See Appendix B for the selection of the network architecture.

TABLE 1 | MSPE (calculated across both space and time) for the forecasted $\mathbf{y}_t$ at all locations and time instances in 2016.

Model	1 h	2 h	3 h	6 h
Persistence	0.335	0.670	0.936	1.411
ESN-Kriging	0.276	0.424	0.537	0.608
AESN-Kriging	0.268	0.391	0.477	0.587
SA-DESN-Kriging	0.266	0.374	0.461	**0.487**
SA-DESN-STAGA	**0.261**	**0.345**	**0.422**	0.488

Note: In bold is the name of our proposed approach and the minimum MSPE values across lead times.

Table 1 presents the Mean Squared Prediction Error (MSPE) for the forecasted values $\boldsymbol{y}_t$ across all locations and time points in 2016. The results demonstrate a clear performance hierarchy among the evaluated models across different forecasting horizons.

The traditional Persistence approach exhibits the highest error values (0.335, 0.670, 0.936, and 1.411 for 1, 2, 3, and 6 h horizons, respectively), establishing a baseline against which we can evaluate more sophisticated models. The basic ESN-Kriging model offers substantial improvement. Further refinements through the AESN-Kriging approach yield additional performance gains, with MSPE values of 0.268, 0.391, 0.477, and 0.587 across the four forecasting horizons. The SA-DESN-Kriging model shows competitive performance, particularly excelling at the 6-hour horizon with the lowest MSPE of 0.487. The most substantial improvements emerge with the implementation of the SA-DESN-STAGA algorithm. This approach demonstrates superior performance across the first three horizons, achieving MSPE values of 0.261, 0.345, and 0.422 for 1, 2, and 3 h forecasts, respectively. These represent error reductions of 22.1%, 48.5%, and 54.9% compared to the baseline Persistence model. For the 6-hour horizon, SA-DESN-STAGA achieves an MSPE of 0.488, nearly matching the best performing SA-DESN-Kriging model.

The SA-DESN-STAGA model consistently outperforms all other approaches for short to short-term forecasting (1–3 h), while maintaining competitive performance for medium-term predictions. The SA-DESN-STAGA model's ability to maintain optimal or near-optimal performance across all forecasting horizons underscores its value as a robust and versatile forecasting

TABLE 2 | Predictive performance comparison in terms of median MSPE (computed across the temporal dimension) and average inferencing (train, forecast) time for different models (IQR in parenthesis) in 2016.

Model	1 h	2 h	3 h	6 h	Train time (min)	Forecast time (min)
Persistence	0.326 (0.090)	0.651 (0.152)	0.912 (0.187)	1.380 (0.240)	**0.0**	**0.4**
ESN-Kriging	0.271 (0.106)	0.393 (0.173)	0.487 (0.239)	0.656 (0.461)	240.3	109.9
AESN-Kriging	0.269 (0.134)	0.374 (0.186)	0.450 (0.255)	0.610 (0.521)	218.1	84.5
SA-DESN-Kriging	0.260 (0.107)	0.343 (0.198)	0.432 (0.265)	0.621 (0.545)	153.7	85.4
SA-DESN-STAGA	**0.246** (0.169)	**0.334** (0.228)	**0.409** (0.299)	**0.549** (0.522)	86.5	2.1

Note: In bold is the name of our proposed approach and the minimum values for MSPE, train time, and forecast time.

solution for wind speed nowcasting applications. In Figure A2, we also show the prediction error against forecast horizon for the competing methods to further showcase the superior skills of SA-DESN-STAGA.

As shown in Table 2 for median MSPE across time (each MSPE value is computed across space; the median is calculated across time), the SA-DESN-STAGA model consistently achieves the best performance across all forecasting horizons. The baseline Persistence model, while computationally efficient (0.0 training time since we do not need to train, 0.4 min of forecasting), exhibits the highest error rates across all horizons with median MSPE values of 0.326, 0.651, 0.912, and 1.380 for 1, 2, 3, and 6 h predictions, respectively.

The ESN-Kriging approach offers moderate improvements over the baseline but requires substantially higher computational resources (240.3 min training time), demonstrating the limitations of basic echo state networks for this forecasting task. The AESN-Kriging model shows better performance, while requiring slightly less training time (218.1 min) than ESN-Kriging. The SA-DESN-Kriging model demonstrates more competitive performance. The proposed SA-DESN-STAGA model emerges as the optimal choice across all forecasting horizons, achieving the lowest median MSPE values of 0.246, 0.334, 0.409, and 0.549 for 1, 2, 3, and 6 h predictions, respectively. Notably, all three models employing the spatial Kriging method to reconstruct the full-dimensional data from knots demand substantial computational time due to intensive matrix operations. Among these, STAGA stands out with the most efficient computational performance, achieving the fastest forecasting time (2.1 min) once the STAGA training is complete.

The SA-DESN-STAGA model's combination of optimal accuracy and computational efficiency makes it the preferred choice for operational wind forecasting applications. The model successfully balances performance requirements with practical computational constraints, delivering state-of-the-art results while maintaining feasible deployment costs.

We also compare the predictive performance of ensemble model with the ground truth over space, which are given in Figures 4 and A1. We witness that the prediction accuracy and stability are inferior around mountainous regions near the Coastal Red Sea, which is expected since the wind speed is more volatile in these areas. The ensemble model performs well in the flat areas, which are more common in Saudi Arabia.

5.1 | Uncertainty Quantification

To quantify the predictive uncertainty of the proposed SA-DESN-STAGA model, we implement a conditional conformal prediction approach called Mondrian conformal prediction (Boström et al. 2021).

The prediction intervals are first calibrated using the ensemble-based intervals on a held-out set. To avoid data leakage and to preserve the chronological structure of the forecasting problem, we use the years 2013–2014 for model fitting and hyperparameter selection. Hyperparameters are selected by the time series cross-validation scheme introduced previously. After the forecasting model and all hyperparameters are fixed, the year 2015 is used only for conformal calibration, while the year 2016 is held out completely and used only for final out-of-sample evaluation.

Let the validation set $\mathcal{T}_{\text{val}}$ be used to select the SA-DESN and STAGA hyperparameters, while the calibration set $\mathcal{T}_{\text{cal}}$ be used only after the forecasting model has been fixed. The test year $\mathcal{T}_{\text{test}}$ is used only once, for the final assessment of coverage.

For each forecast origin t, spatial location $\mathbf{s}_i$, and forecast horizon $h \in \{1, 2, 3\}$, the ensemble version of SA-DESN-STAGA generates C forecasts $\hat{Y}^{(1)}_{t+h}(\mathbf{s}_i), \ldots, \hat{Y}^{(C)}_{t+h}(\mathbf{s}_i)$, for the wind residual $Y_{t+h}(\mathbf{s}_i)$. For a nominal prediction level $1 - \alpha$, we first construct the uncalibrated central ensemble interval

$$\hat{I}^{\,0}_{t,h,i}(\alpha) = \left[\hat{Q}_{\alpha/2,t,h,i}, \hat{Q}_{1-\alpha/2,t,h,i}\right],$$

where $\hat{Q}_{p,t,h,i}$ denotes the empirical p-th quantile of the ensemble forecasts $\{\hat{Y}^{(c)}_{t+h}(\mathbf{s}_i)\}_{c=1}^{C}$. We then compute the conformal nonconformity score on the calibration set as

$$R^{(\alpha)}_{t,h,i} = \max\left\{\hat{Q}_{\alpha/2,t,h,i} - Y_{t+h}(\mathbf{s}_i), Y_{t+h}(\mathbf{s}_i) - \hat{Q}_{1-\alpha/2,t,h,i}, 0\right\}.$$

This score is zero when the observation is already contained in the uncalibrated ensemble interval and is positive when the interval must be expanded to include the observation.

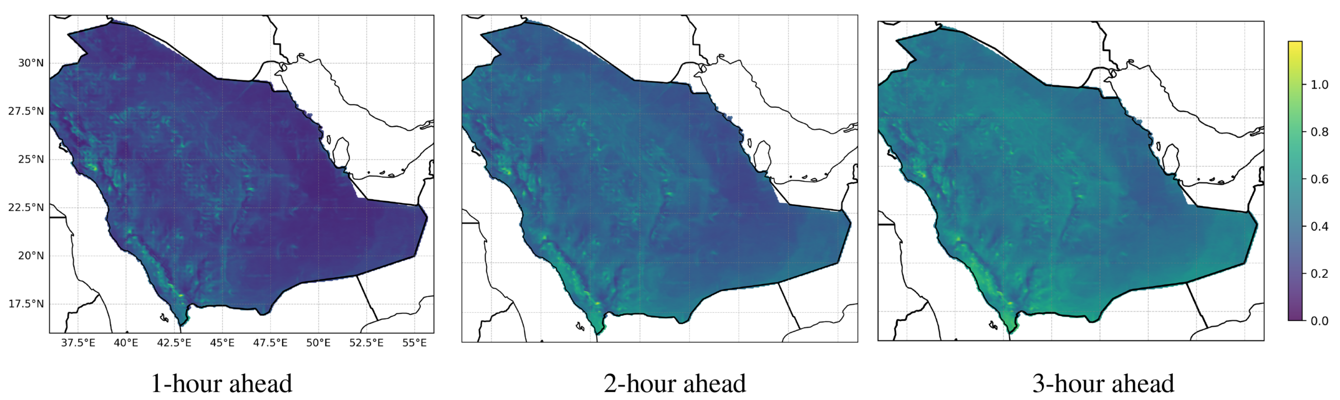

FIGURE 4 | Wind speed MSPE using SA-DESN-STAGA model at all locations for different forecast horizons. (a) 1-hour ahead, (b) 2-hour ahead, (c) 3-hour ahead.

TABLE 3 | The median proportion of wind speed residuals $\mathbf{y}_t$ in 2016 falling into the associated Mondrian conformal prediction intervals.

Prediction interval	1-hour ahead	2-hour ahead	3-hour ahead
95%	95.6% (1.9%)	95.2% (2.1%)	94.7% (2.6%)
80%	81.0% (2.7%)	80.5% (3.0%)	79.8% (3.5%)
60%	60.9% (2.5%)	60.4% (2.9%)	59.8% (3.4%)

Note: Interquartile ranges across all spatial locations are shown in parentheses.

Because the previous undercoverage was strongest for longer lead times and because wind-speed variability is seasonal, we use Mondrian classes indexed jointly by forecast horizon and calendar month. Specifically, we define the Mondrian group label $G(t,h) = (h, M(t+h))$, where $M(t+h) \in \{1, \dots, 12\}$ is the calendar month of the target time $t+h$. For each location $\mathbf{s}_i$, nominal level $1-\alpha$, and Mondrian class g, define the calibration score set

$$\mathcal{R}_{i,g}^{(\alpha)} = \left\{ R_{t,h,i}^{(\alpha)} : t \in \mathcal{T}_{\text{cal}},\ G(t,h) = g \right\}.$$

Let $n_{i,g}$ denote the number of scores in $\mathcal{R}_{i,g}^{(\alpha)}$. The conformal correction is then the finite-sample conformal quantile

$$\hat{q}_{i,g}^{(\alpha)} = \text{Quantile}_{\lceil (n_{i,g}+1)(1-\alpha) \rceil / n_{i,g}} \left(\mathcal{R}_{i,g}^{(\alpha)} \right).$$

The final Mondrian conformal prediction interval for $Y_{t+h}(\mathbf{s}_i)$ is

$$\hat{I}_{t,h,i}(\alpha) = \left[\hat{Q}_{\alpha/2,t,h,i} - \hat{q}_{i,G(t,h)}^{(\alpha)},\ \hat{Q}_{1-\alpha/2,t,h,i} + \hat{q}_{i,G(t,h)}^{(\alpha)} \right].$$

This construction has two practical advantages. First, the correction is horizon-specific, so the three-hour-ahead intervals are allowed to be wider than the one-hour-ahead intervals when the calibration residuals indicate stronger error accumulation. Second, the correction is month-specific, so the conformal interval adapts to seasonal changes in wind variability rather than applying a single global correction across the entire year.

Table 3 reports the empirical coverage of the Mondrian conformal prediction intervals for the 2016 test year. The entries are the median coverage proportions across all spatial locations, with interquartile ranges in parentheses.

Small deviations from nominal levels can be attributed to two factors. First, although the calibration split is seasonally balanced, the calibration and test years are not strictly exchangeable because the data form a temporally dependent spatio-temporal process. Second, the three-hour forecasts are obtained recursively from shorter-lead predictions, so errors can accumulate with lead time. The Mondrian conformal correction directly addresses these effects by calibrating separately across forecast horizons and calendar months, leading to more reliable prediction intervals for operational nowcasting.

6 | Wind Energy Forecasts in Saudi Arabia

This section explores the practical utility of our forecasting approach by converting wind speed predictions into wind power estimates and analyzing their operational advantages. As Saudi Arabia currently lacks an established energy market with time-dependent pricing structures, we focus our analysis on the two-hour ahead wind power forecasts as a reference point.

TABLE 4 | The annual sum of the absolute differences in wind energy for all the selected 75 wind turbines in Saudi Arabia.

Model	Sum of absolute differences
Persistence	3.12×10^8 kWh
ESN-Kriging	2.77×10^8 kWh
AESN-Kriging	2.67×10^8 kWh
SA-DESN-Kriging	2.64×10^8 kWh
SA-DESN-STAGA	2.61×10^8 kWh

Converting surface-level wind speeds to turbine hub height requires vertical extrapolation, as the WRF simulations provide data at 10 meters while commercial wind turbines typically operate at heights between 80 and 120 meters (Crippa et al. 2021). We employ the widely-accepted power law relation (Tagle et al. 2019):

$$Z_t^{(h)}(\mathbf{s}) = Z_t(\mathbf{s}) \left(\frac{h}{10} \right)^{\alpha(\mathbf{s})} \exp(\varepsilon_t),$$

where h represents the hub height, $\alpha(\mathbf{s})$ denotes the location-specific wind shear coefficient governing the extrapolation rate, and $\varepsilon_t \sim N(0, \sigma^2)$ accounts for random error that is independently and identically distributed across time and space.

Our analysis concentrates on 75 optimal inland locations for wind farm deployment identified by Giani et al. (2020). These sites were selected through comprehensive optimization considering multiple factors including construction and maintenance costs, power line installation requirements, and energy grid integration challenges (Wang et al. 2026). For each location, we utilize the power curve of the specific turbine model determined to be optimal for that site.

The annual sum of total absolute differences across all locations is calculated and presented in Table 4, providing a quantitative assessment of the operational benefits achieved through our forecasting methodology. The SA-DESN-STAGA model demonstrates superior performance with the lowest cumulative error (2.61×10^8 kWh).

Figure 5 presents the performance evaluation of the SA-DESN-STAGA wind energy forecasting model across different quantiles during 2016 at 75 wind turbines. The left panel displays the annual performance on a logarithmic scale, revealing a characteristic U-shaped pattern where forecast accuracy is highest around the median (55th to 65th percentile) with a minimum logarithmic sum of absolute differences around 19.4, while accuracy deteriorates toward the extreme quantiles. This U-shaped distribution indicates that the model performs optimally for central tendency predictions but faces increased uncertainty when forecasting extreme wind energy events. The right panel shows the monthly performance variation, where the magenta shaded region represents the confidence interval around the median forecast performance. The monthly analysis demonstrates relatively

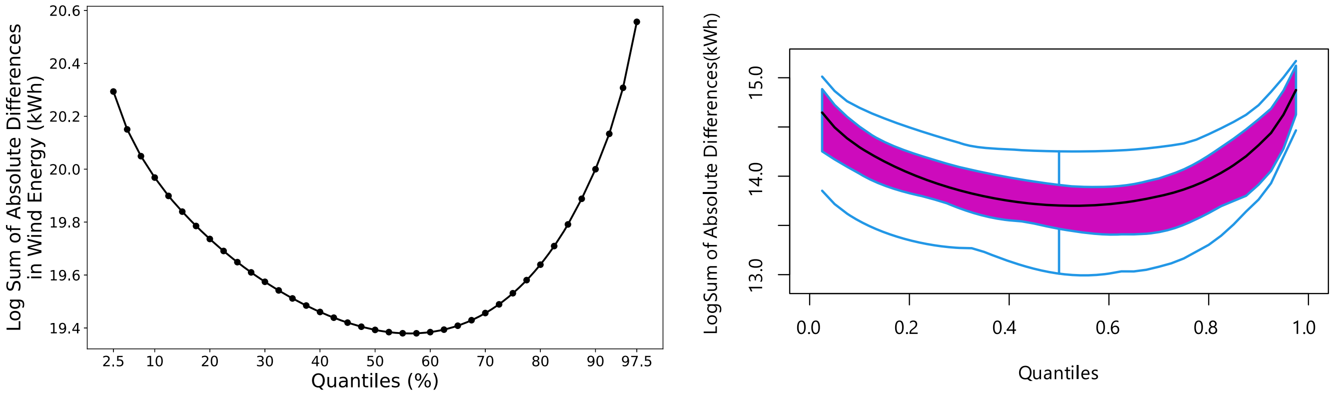

FIGURE 5 | *Left:* logarithm scale of annual sum of the absolute differences between SA-DESN-STAGA forecast quantiles and the truth in wind energy during 2016 at all the 75 wind turbines combined. *Right:* logarithm scale of the monthly sum of the absolute differences between SA-DESN-STAGA forecast quantiles and the ground truth in wind energy during 2016 at all the 75 wind turbines combined.

stable performance, suggesting that the SA-DESN-STAGA model provides consistent forecasting accuracy across different temporal scales while exhibiting the expected challenges associated with extreme quantile predictions.

6.1 | Regional and Seasonal Performance

To assess practical utility across Saudi Arabia's diverse geography, we partition the 75 wind farm locations into six climate regions following Almazroui et al. (2014): An Nafud, Coastal Red Sea, Najd Interior, Asir Highland, West Rub al-Khali, and East Rub al-Khali (Figure A3). Table 5 reports the region-averaged annual Sum of Absolute Differences (SAD) and the seasonal aggregated SAD across all 75 farms, providing both a spatially normalized and a temporally stratified view of prediction accuracy.

6.1.1 | Regional Analysis (Panel A)

The region-averaged annual SAD reveals meaningful spatial heterogeneity in model performance. The **Asir Highland** region exhibits the largest averaged error (4.26×10^6 MWh, 30.8% of the cross-regional sum), consistent with the complex orographic wind patterns and sharp elevation gradients characteristic of the Asir mountain range—features that are inherently difficult to capture with any statistical or deep-learning model. The **Najd Interior** follows closely (3.86×10^6 MWh, 27.9%), reflecting the high inter-annual variability of continental plateau winds. The **An Nafud** desert region shows a moderate error (3.60×10^6 MWh, 26.0%), attributable to the episodic and directionally variable shamal wind events. Notably, the **Coastal Red Sea** region achieves the smallest averaged error (2.16×10^6 MWh, 15.6%), suggesting that the relatively steady sea-breeze circulation along the coast produces a more stationary and learnable wind regime. Averaging by the number of farms per region is essential here, as a naive summation would conflate regional predictability with wind farm density.

6.1.2 | Seasonal Analysis (Panel B)

The seasonal SAD results reveal a clear warm-season amplification of prediction error. **Summer (JJA)** contributes the highest share of annual error (7.36×10^7 MWh, 28.2%), followed by **Spring (MAM)** (7.15×10^7 MWh, 27.4%). Both seasons coincide with the intensified shamal wind system and increased convective activity over the Arabian Peninsula, generating large-amplitude, rapidly fluctuating wind speed events that challenge the model's temporal extrapolation. In contrast, **Winter (DJF)** records the lowest seasonal error (5.67×10^7 MWh, 21.7%), consistent with the more stable synoptic-scale circulation patterns during the boreal winter. **Autumn (SON)** occupies an intermediate position (5.93×10^7 MWh, 22.7%), representing a transitional regime between the stable winter and the volatile spring–summer period. The Summer–Winter contrast ($\sim$ 30%

TABLE 5 | Summary of wind power prediction errors by climate region and season.

Panel	Stratum	SAD (kWh)	Relative share (%)
(A) Regional breakdown—region-averaged annual SAD			
	An Nafud	3.600×10^6	26.0
	Coastal Red Sea	2.160×10^6	15.6
	Najd Interior	3.860×10^6	27.9
	Asir Highland	4.260×10^6	30.8
	Overall average	$\mathbf{3.470 \times 10^6}$	—
(B) Seasonal breakdown—total aggregated SAD across all farms			
	Winter	5.669×10^7	21.7
	Spring	7.150×10^7	27.4
	Summer	7.364×10^7	28.2
	Autumn	5.934×10^7	22.7
	Annual total	$\mathbf{2.612 \times 10^8}$	—

Note: Upper panel: Region-averaged annual Sum of Absolute Differences (SAD, in kWh), computed as the total annual absolute difference divided by the number of wind farms allocated to each climate region, to account for the uneven distribution of the 75 wind farms across regions. *Lower panel*: Seasonal aggregated SAD (in kWh) summed across all 75 wind farms and all time steps within each season (DJF = Winter, MAM = Spring, JJA = Summer, SON = Autumn). All values are rounded to four significant figures.

relative increase) underscores the need for season-aware modeling strategies in future work.

6.1.3 | Implications for Vision 2030

Saudi Arabia's Vision 2030 plan targets 16 GW of wind power capacity by 2030, with projects concentrated along the Coastal Red Sea corridor and the northern An Nafud region (Nurunnabi 2017). Our results show that SA-DESN-STAGA achieves its lowest prediction errors in precisely these two regions, supporting its suitability for operational deployment at the sites of highest policy priority. The model's short-horizon accuracy (1–3 h ahead) is particularly relevant for real-time grid dispatch. The higher errors observed in the Asir Highland region are consistent with the known orographic complexity there and with current plans that favour lower-elevation coastal and desert sites for utility-scale wind development. These findings suggest that SA-DESN-STAGA offers a practically viable forecasting tool for the wind energy scenarios most critical to Saudi Arabia's renewable energy transition.

7 | Conclusion

In this study, we proposed a novel framework for wind speed forecasting that integrates spatial information with deep echo state networks (SA-DESN) and graph attention autoencoders (STAGA). Our approach leverages the strengths of both echo state networks and autoencoders to effectively capture the complex spatio-temporal dynamics of wind speed data. We demonstrated the effectiveness of our method through extensive experiments on simulated wind speed datasets, achieving significant improvements in forecasting accuracy compared to traditional methods.

The proposed framework not only enhances the accuracy of wind speed predictions but also provides a robust solution for uncertainty quantification, enabling the conformal prediction intervals as requested in machine learning community (Xu and Xie 2023). Additionally, we explored the practical implications of our forecasting approach by converting wind speed predictions into wind power estimates, highlighting its operational advantages in real-world applications.

The results indicate that our proposed model, which actively combines spatial information within SA-DESN, consistently outperforms individual models across various forecasting horizons.

Future research will aim to improve the model's capabilities by investigating additional mechanisms and integrating advanced deep learning components like Transformers (Vaswani et al. 2017). Additionally, the current uncertainty quantification results do not fully align with the expected coverage rates. To address this, we plan to examine the potential of conformal prediction methods (Vovk et al. 2005) to enhance the reliability of prediction intervals.

Acknowledgments

This publication is based upon work supported by King Abdullah University of Science and Technology Research Funding (KRF) under Award No. ORFS-2022-CRG11-5069.

Funding

This work was supported by King Abdullah University of Science and Technology (Grant No. ORFS-2022-CRG11-5069).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Akpinar, E. K., and S. Akpinar. 2005. "An Assessment on Seasonal Analysis of Wind Energy Characteristics and Wind Turbine Characteristics." *Energy Conversion and Management* 46, no. 11–12: 1848–1867.

Almazroui, M., R. Dambul, M. N. Islam, and P. D. Jones. 2014. "Principal Components-Based Regionalization of the Saudi Arabian Climate." *International Journal of Climatology* 35, no. 9: 25552573. https://doi.org/10.1002/joc.4139.

Blanco, M. I. 2009. "The Economics of Wind Energy." *Renewable and Sustainable Energy Reviews* 13, no. 6–7: 1372–1382.

Bonas, M., and S. Castruccio. 2023. "Calibration of Spatiotemporal Forecasts From Citizen Science Urban Air Pollution Data With Sparse Recurrent Neural Networks." *Annals of Applied Statistics* 17, no. 3: 1820–1840.

Bonas, M., and S. Castruccio. 2025. "High-Resolution Urban Air Quality Monitoring From Citizen Science Data With Echo-State Transformer Networks." *Journal of the Royal Statistical Society Series C: Applied Statistics* 74, no. 4: 905–924. https://doi.org/10.1093/jrsssc/qlaf007.

Boström, H., U. Johansson, and T. Löfström. 2021. "Mondrian Conformal Predictive Distributions." In *Proceedings of the Tenth Symposium on Conformal and Probabilistic Prediction and Applications*, edited by L. Carlsson, Z. Luo, G. Cherubin, and K. A. Nguyen, vol. 152, 24–38. PMLR. https://proceedings.mlr.press/v152/bostrom21a.html.

Cassola, F., and M. Burlando. 2012. "Wind Speed and Wind Energy Forecast Through Kalman Filtering of Numerical Weather Prediction Model Output." *Applied Energy* 99: 154–166.

Chkeir, S., A. Anesiadou, A. Mascitelli, and R. Biondi. 2023. "Nowcasting Extreme Rain and Extreme Wind Speed With Machine Learning Techniques Applied to Different Input Datasets." *Atmospheric Research* 282: 106548.

Crippa, P., M. Alifa, D. Bolster, M. G. Genton, and S. Castruccio. 2021. "A Temporal Model for Vertical Extrapolation of Wind Speed and Wind Energy Assessment." *Applied Energy* 301: 117378.

Deetz, C. H. 1918. *Cartography: The Lambert Conformal Conic Projection With Two Standard Parallels Including a Comparison of the Lambert Projection With the Bonne and Polyconic Projections*. Vol. 4. US Government Printing Office.

Farag, A. A. 2019. "The Story of Neom City: Opportunities and Challenges." In *New Cities and Community Extensions in Egypt and the Middle East: Visions and Challenges*, 35–49. Springer.

Gao, H., and S. Ji. 2019. "Graph U-Nets." In *International Conference on Machine Learning*, 2083–2092. PMLR.

Giani, P., F. Tagle, M. G. Genton, S. Castruccio, and P. Crippa. 2020. "Closing the Gap Between Wind Energy Targets and Implementation for Emerging Countries." *Applied Energy* 269: 115085.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9, no. 8: 1735–1780.

Huang, H., S. Castruccio, and M. G. Genton. 2022. "Forecasting High-Frequency Spatio-Temporal Wind Power With Dimensionally

Reduced Echo State Networks." *Journal of the Royal Statistical Society Series C: Applied Statistics* 71, no. 2: 449–466.

Jaeger, H. 2007. "Echo State Network." *Scholarpedia* 2, no. 9: 2330.

Janić, Z. I. 2001. "Nonsingular Implementation of the Mellor-Yamada Level 2.5 Scheme in the Ncep Meso Model."

Kinley, R. 2017. "Climate Change After Paris: From Turning Point to Transformation." *Climate Policy* 17, no. 1: 9–15.

Kipf, T. N., and M. Welling. 2016. "Semi-Supervised Classification With Graph Convolutional Networks." *arXiv Preprint arXiv:1609.02907*.

Krishna, V. B., W. S. Wadman, and Y. Kim. 2018. "Nowcasting: Accurate and Precise Short-Term Wind Power Prediction Using Hyperlocal Wind Forecasts." In *Proceedings of the Ninth International Conference on Future Energy Systems*, 63–74. Association for Computing Machinery.

Liu, C., R. Yao, L. Zhang, and Y. Liao. 2019. "Attention Based Echo State Network: A Novel Approach for Fault Prognosis." In *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, 489–493. Association for Computing Machinery.

McDermott, P. L., and C. K. Wikle. 2018. "Deep Echo State Networks With Uncertainty Quantification for Spatio-Temporal Forecasting." *Environmetrics* 30, no. 3. https://doi.org/10.1002/env.2553.

Medsker, L. R., and L. Jain. 2001. "Recurrent Neural Networks." *Design and Applications* 5, no. 64–67: 2.

Molteni, F., R. Buizza, T. N. Palmer, and T. Petroliagis. 1996. "The Ecmwf Ensemble Prediction System: Methodology and Validation." *Quarterly Journal of the Royal Meteorological Society* 122, no. 529: 73–119.

Nakajima, K., and I. Fischer. 2021. *Reservoir Computing*. Springer.

Nurunnabi, M. 2017. "Transformation From an Oil-Based Economy to a Knowledge-Based Economy in Saudi Arabia: The Direction of Saudi Vision 2030." *Journal of the Knowledge Economy* 8: 536–564.

Shi, X., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. 2015. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting." In *Proceedings of the 29th International Conference on Neural Information Processing Systems*, 802–810. MIT Press.

Shumway, R. H., and D. S. Stoffer. 2017. "ARIMA Models." *Time Series Analysis and Its Applications: With R Examples*, 85–175. Springer Nature Switzerland.

Spera, D. A. 1994. "Wind Turbine Technology."

Tagle, F., S. Castruccio, P. Crippa, and M. G. Genton. 2019. "A Non-Gaussian Spatio-Temporal Model for Daily Wind Speeds Based on a Multi-Variate Skew-t Distribution." *Journal of Time Series Analysis* 40, no. 3: 312–326.

Tschannen, M., O. Bachem, and M. Lucic. 2018. "Recent Advances in Autoencoder-Based Representation Learning." *arXiv Preprint arXiv:1812.05069*.

Vaswani, A., N. Shazeer, N. Parmar, et al. 2017. "Attention Is All You Need." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. MIT Press.

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. 2018. "Graph Attention Networks." In *International Conference on Learning Representations (ICLR)*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ.

Vovk, V., A. Gammerman, and G. Shafer. 2005. *Algorithmic Learning in a Random World*. Vol. 29. Springer.

Wang, K., M. Kim, S. Castruccio, and M. G. Genton. 2026. "Modeling High-Resolution Spatio-Temporal Wind With Deep Echo State Networks and Stochastic Partial Differential Equations." *Journal of the Royal Statistical Society. Series C, Applied Statistics* 75, no. 2: 303–319.

Wang, Y., M. Long, J. Wang, Z. Gao, and P. S. Yu. 2017. "PredRNN: Recurrent Neural Networks for Predictive Learning Using Spatiotemporal LSTMs." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 879–888. MIT Press.

Wang, Z., S. H. Holan, and C. K. Wikle. 2025. "Echo State Networks for Spatio-Temporal Area-Level Data." *Data Science in Science* 4, no. 1: 2554883. https://doi.org/10.1080/26941899.2025.2554883.

Werbos, P. J. 1990. "Backpropagation Through Time: What It Does and How to Do It." *Proceedings of the IEEE* 78, no. 10: 1550–1560.

Xu, C., and Y. Xie. 2023. "Conformal Prediction for Time Series." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, no. 10: 11575–11587.

Yildiz, I. B., H. Jaeger, and S. J. Kiebel. 2012. "Re-Visiting the Echo State Property." *Neural Networks* 35: 1–9.

Zhang, S., H. Tong, J. Xu, and R. Maciejewski. 2019. "Graph Convolutional Networks: A Comprehensive Review." *Computational Social Networks* 6, no. 1: 1–23.

Zhu, X., and M. G. Genton. 2012. "Short-Term Wind Speed Forecasting for Power System Operations." *International Statistical Review* 80, no. 1: 2–23.

Appendix A: Additional Figures

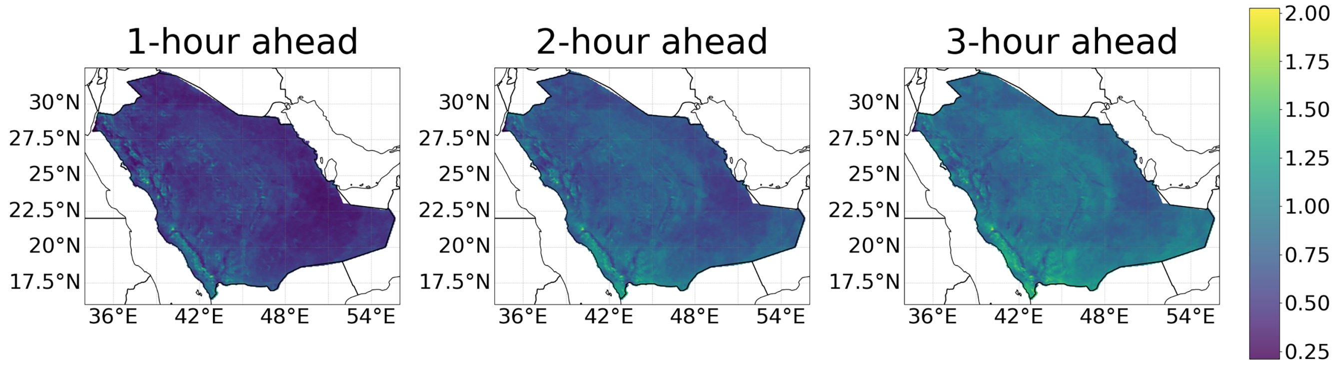

FIGURE A1 | Standard deviation of squared prediction error (calculated across time) over 1h to 3h horizons.

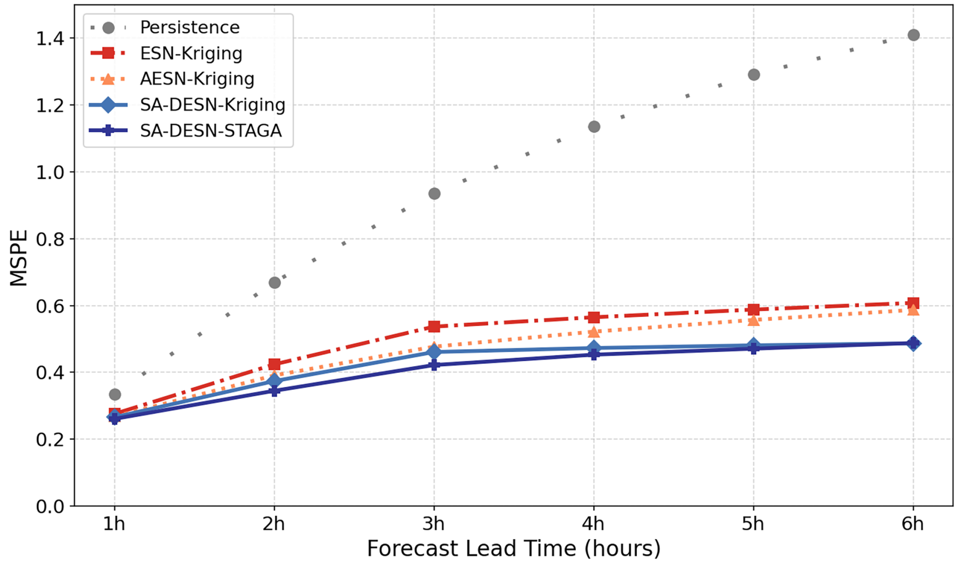

FIGURE A2 | Time horizons vs. MSPE.

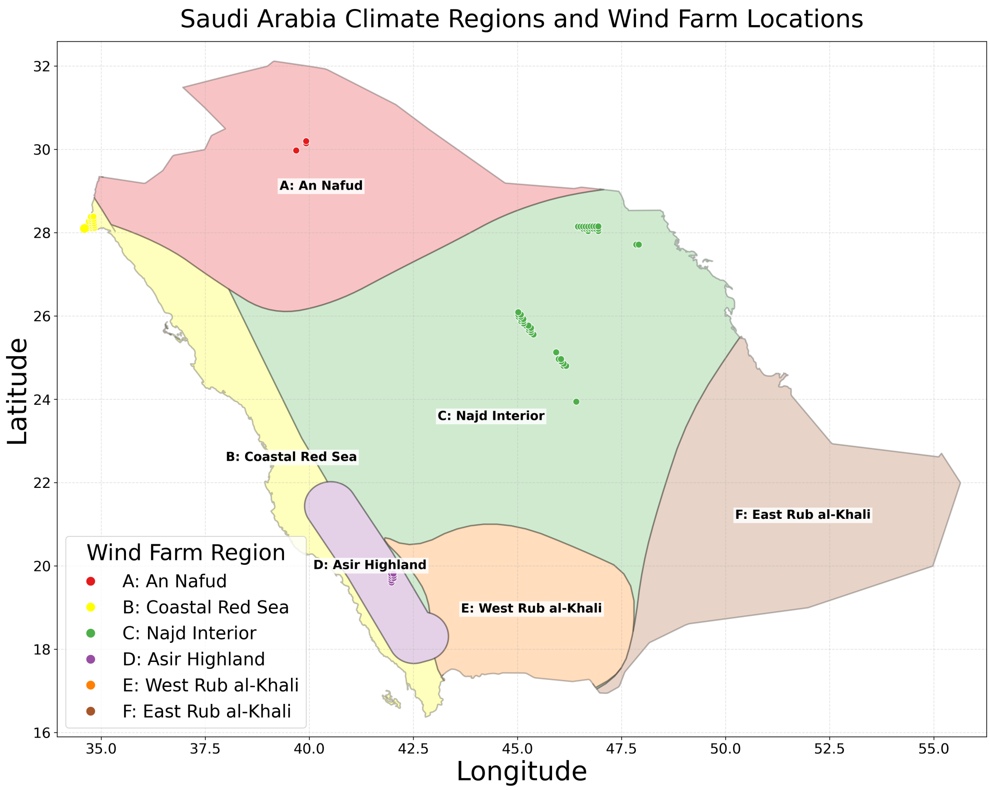

FIGURE A3 | Locations of the 75 wind farms overlaid on the identified climate regions. The climate regions are adapted from Almazroui et al. (2014). Each point denotes a wind farm, with colors indicating the corresponding climate region classification. One can notice that there are no wind farms in West and East Rub al-Khali.

Appendix B: Selection of Network Architecture

The choice of the latent compression dimension $n = 2500$ in STAGA and the reservoir depth D in SA-DESN warrants empirical justification. We therefore conduct two sensitivity studies, holding all other settings fixed. Throughout this section, forecast skill is measured by the MSPE of $\mathbf{y}_t$ on 2016, and reconstruction skill by the RMSE of the STAGA autoencoder on a held-out subset of 2015. Reported numbers are averages over three random seeds.

Sensitivity to the Latent Dimension n

We vary $n \in \{500, 1000, 1500, 2000, 2500, 3000, 4000\}$ with all other hyperparameters fixed at their defaults ($D = 2$, k-NN scheme unchanged). As shown in Table B1, increasing n reduces reconstruction error and MSPE consistently up to $n = 2500$, beyond which the accuracy gains become negligible while training time grows by over 50%. We therefore select $n = \mathbf{2500}$ as the operating point.

TABLE B1 | Sensitivity of forecast accuracy and reconstruction error to the STAGA latent dimension n.

n	RMSE$_{\text{rec}}$	MSPE@1h	MSPE@2h	MSPE@3h	Time (min)
500	0.85	0.412	0.501	0.594	**2.1**
1000	0.52	0.328	0.418	0.498	4.6
1500	0.38	0.289	0.376	0.452	7.3
2000	0.31	0.272	0.357	0.434	11.3
2500	0.27	**0.261**	**0.345**	**0.422**	16.0
3000	0.26	0.263	0.348	0.425	22.9
4000	**0.25**	0.268	0.355	0.433	34.6

Note: $D = 2$ and all other hyperparameters are fixed. Best value per column in bold.

Sensitivity to the SA-DESN Depth D

With $n = 2500$ fixed, we vary $D \in \{1, 2, 3\}$. Table B2 shows that accuracy improves from $D = 1$ to $D = 2$, where the two-layer reservoir captures the dominant multi-scale temporal dependence of the latent residuals. Adding a third layer slightly *degrades* performance across all horizons, consistent with over-contraction of the echo-state dynamics reducing sensitivity to recent inputs, while also increasing training time by an additional 75% over $D = 2$. We therefore select $D = \mathbf{2}$ as the default depth.

The final configuration $n = 2500$, $D = 2$ balances forecasting accuracy and computational cost, and is used throughout the main experiments.

TABLE B2 | Sensitivity of forecast accuracy to the SA-DESN depth D at $n = 2500$.

D	MSPE@1h	MSPE@2h	MSPE@3h	Time (min)
1	0.318	0.402	0.477	**5.7**
2	**0.261**	**0.345**	**0.422**	16.0
3	0.269	0.353	0.431	34.4

Note: Best value per column in bold.

Appendix C: Notation

We provide a summary of the key notations used throughout this article, along with their definitions in Table C1.

TABLE C1 | Summary of notations.

Notation	Description
$\mathbb{R}^{m\times n}$	Real matrix space of dimension $m \times n$
N	Number of spayial locations
m	Number of time lags
n	Reduced number of spatial locations
T	Number of time steps
$Y_t(\mathbf{s}_i)$	Residual wind speed at location $\mathbf{s}_i$ at time t
$[\cdot,\cdot]$	Concatenation operator
$\mathbf{v}[:]$	Extracting all elements within vector $\mathbf{v}$
$\|\cdot\|_F$	Frobenius norm
$U(a,b)$	Uniform distribution and parameters a and b
$\text{Bern}(p)$	Bernoulli distribution with parameter p
$I(\cdot)$	Indicator function

Appendix D: Areal Representation for ESN

Wang et al. (2025) proposed capturing spatial dynamics at the input stage, offering two key advantages: (1) preserving desirable echo state property, and (2) minimizing computational overhead by avoiding recursive computations at each time step.

Their approach, termed Areal Echo State Network (AESN), uses random representations to capture spatial dynamics. The areal random representation transforms input signals into graph spectral domains using

$$\mathbf{S} = \widetilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2}, \tag{D1}$$

$$\mathbf{z}^{(l)} = \mathbf{S}(\mathbf{U}^{(l)} \odot \mathbf{X}\cdot\mathbf{1}), \tag{D2}$$

$$\mathbf{U}^{(l)} \sim \mathrm{U}(-a_u, a_u), \tag{D3}$$

where $\odot$ is the Hadamard product, $\mathbf{U}^{(l)}$ are randomly sampled weights, $\mathbf{S}$ is the symmetric normalized adjacency matrix, and $\widetilde{\mathbf{D}}$ represents the degree matrix of $\widetilde{\mathbf{A}}$. Here, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the proximity matrix and $\widetilde{\mathbf{A}} = \mathbf{I} + \mathbf{A}$ is the proximity matrix with self-loops included. All definitions follow the same logic as defined in Kipf and Welling (2016). Unlike traditional GCNs (Zhang et al. 2019), the AESN method uses independent random weights sampled from a uniform distribution rather than learned weights.

The AESN integrates embedded signals by

$$\tilde{\mathbf{h}}_t = g_h\left(\frac{\nu}{|\lambda_w|}\mathbf{W}_{res}\mathbf{h}_{t-1} + \mathbf{W}_{in}\tilde{\mathbf{z}}_t\right), \tag{D4}$$

where $\tilde{\mathbf{z}}_t = \text{vec}(\mathbf{Z}_t)$ and $\mathbf{Z}_t \in \mathbb{R}^{N\times L}$ is the embedding matrix, which is a collection of $\{\mathbf{z}_t^{(1)}, \ldots, \mathbf{z}_t^{(L)}\}$.

Remark 1. The AESN method is a special case of the ESN model, where only the input layer is replaced by the areal random representation. This approach allows for the integration of spatial information into the ESN framework while maintaining the advantages of the echo state property and computational efficiency.