Electric Power Systems Research 260 (2026) 113259

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Decomposition-driven Mamba state space models with expert routing for wind curtailment forecasting

Laio Oriel Seman [a], Stefano Frizzo Stefenon [b,*], João Pedro Matos-Carvalho [c]

[a] Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, Brazil
[b] Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Rua Conselheiro Emídio Navarro 1, 1959-007, Lisboa, Portugal
[c] LASIGE, Faculdade de Ciências, Universidade de Lisboa, Lisboa, 1749-016, Portugal

ARTICLE INFO

Keywords:
Mamba architecture
Mixture-of-experts
Reversible instance normalization
Variational mode decomposition
Wind curtailment

ABSTRACT

Wind power curtailment, known in Brazil as constrained-off, has intensified with the rapid expansion of renewable generation, driven by transmission operational constraints. Accurate forecasting of curtailment is essential for grid planning, compensation mechanisms, and infrastructure optimization. However, constrained-off time series exhibit strong non-stationarity, multi-scale behavior, and long-range dependencies that challenge conventional forecasting models. This paper proposes a decomposition-driven Mamba state space framework with Mixture-of-Experts (MoE) routing for wind curtailment forecasting. The approach integrates Entropy-Guided Variational Mode Decomposition (EG-VMD) for adaptive trend and noise separation, Reversible Instance Normalization to mitigate distribution shift, multi-scale seasonal-trend modeling, and a sparse expert-based Mamba backbone with linear-time complexity. In benchmark comparisons against state-of-the-art forecasting models, the proposed MoE-Mamba reduces Mean Squared Error (MSE) by up to 27%, Root Mean Squared Error by 14.6%, and Mean Absolute Error by 16.1% relative to the strongest baselines, while improving upon a single-expert Mamba by 15.6% in MSE. The results highlight the effectiveness of decomposition-enhanced State Space Models with expert routing for scalable, long-horizon forecasting in renewable power systems.

1. Introduction

The rapid expansion of renewable energy sources, particularly wind power, has transformed the global energy landscape, offering sustainable alternatives to fossil fuels while addressing climate change imperatives [1]. In Brazil, wind generation has experienced explosive growth, especially in the Northeast region, where favorable wind regimes have driven installed capacity to exceed 30 GW [2]. This surge has positioned Brazil as a leader in renewable integration within Latin America, with wind contributing over 15% to the national electricity mix during peak seasons. However, this progress is accompanied by significant challenges, including grid congestion, transmission bottlenecks, and operational constraints that lead to wind power curtailment, known locally as constrained-off events.

Constrained-off occurs when wind farms are instructed by the National Electric System Operator (ONS) to reduce or halt generation despite available wind resources and plant readiness. These events stem from system-level factors such as transmission overloads, voltage stability issues, security criteria, or energy surpluses, rather than local plant failures [3]. In high-penetration renewable systems like Brazil's National Interconnected System (SIN), curtailment rates have escalated, with projections indicating national averages of 8–11% by 2035 and occasional monthly peaks exceeding 10% [4]. Such curtailments impose substantial economic burdens on generators through lost revenue and uncompensated opportunity costs, while undermining grid efficiency and renewable utilization goals. Accurate forecasting of constrained-off volumes and events is thus critical for enabling proactive operational planning, optimizing storage and dispatch strategies, informing compensation mechanisms, and guiding infrastructure investments to mitigate integration barriers [5].

Traditional forecasting approaches, including statistical models and early machine learning methods, have shown limitations in capturing the volatile, multivariate, and long-range dependencies inherent to curtailment time series [6]. Recent advancements in deep learning, particularly Transformer-based architectures, have improved performance but often suffer from quadratic computational complexity, making them inefficient for long-sequence tasks [7]. State Space Models (SSMs), exemplified by the Mamba architecture, offer a promising alternative with

* Corresponding author.
E-mail addresses: laio@gos.ufsc.br (L.O. Seman), stefano.stefenon@isel.pt (S.F. Stefenon), jpecarvalho@ciencias.ulisboa.pt (J.P. Matos-Carvalho).

https://doi.org/10.1016/j.epsr.2026.113259
Received 16 February 2026; Received in revised form 1 April 2026; Accepted 2 May 2026
Available online 19 May 2026

linear-time scaling and selective state spaces that enable content-aware reasoning [8]. Adaptations like `S-Mamba`, `DMamba`, and `DTMamba` have demonstrated superior efficiency and accuracy in general time series forecasting [9], while power-specific variants such as `PowerMamba` and `DPFMformer` highlight their potential for renewable applications [10].

This work introduces a tailored Mamba-based framework for constrained-off forecasting, leveraging enhanced decomposition strategies and channel interaction modeling to address the unique characteristics of curtailment data. We utilize the ONS open dataset on wind farm operational restrictions, spanning October 2023 to February 2026, which provides detailed monthly records of constrained-off events for Tipo I, II-B, and II-C modalities [11]. Our contributions include: (i) a domain-adapted Mamba architecture that outperforms benchmarks in long-horizon prediction; (ii) an ablation study elucidating the roles of tokenization modes, channel mixing, decomposition, and normalization; and (iii) validation based on measured ONS data from October 2023 to February 2026, demonstrating reduced Mean Squared Error (MSE) and Mean Absolute Error (MAE) compared to state-of-the-art models.

The proposed MoE-Mamba model provides computational advantages over Transformer-based architectures for long-horizon forecasting due to its linear time complexity and conditional computation mechanism. While Transformers rely on self-attention operations with quadratic complexity with respect to the sequence length, the Mamba state space backbone scales linearly, making it more suitable for long sequences and large forecasting horizons. In addition, the MoE routing strategy further improves efficiency by activating only a subset of specialized experts for each input, reducing unnecessary computations while increasing model capacity. This design enables the proposed framework to achieve competitive or superior predictive performance while maintaining lower computational cost and better scalability for long-horizon time series forecasting tasks. These characteristics reinforce the practical innovation of the MoE-Mamba architecture in comparison with conventional Transformer-based approaches.

2. Related works

Recent advancements in sequence modeling have shifted toward SSMs as efficient alternatives to Transformer-based architectures, particularly for handling long sequences with linear computational complexity. The foundational Mamba model introduced selective state spaces, enabling content-aware reasoning via input-dependent parameter selection and achieving state-of-the-art performance across language, audio, and other domains while scaling linearly with sequence length [8]. This efficiency has spurred adaptations to time series forecasting, where capturing long-range temporal dependencies is critical for high predictive accuracy.

Several works have explored and refined Mamba for time series tasks. `S-Mamba` proposes a lightweight variant with linear tokenization per variate and standard Mamba blocks, demonstrating superior accuracy and efficiency over Transformer baselines on benchmark datasets [12]. To address heterogeneous time series components, `DMamba` incorporates explicit seasonal-trend decomposition aligned with Mamba's selective mechanism, using differentially complex modules (bidirectional Mamba for seasonal and Multilayer Perceptron (MLP) for trend) for improved long-range dependency modeling [13]. `DTMamba` employs innovative dual twin Mamba blocks with residual connections to extract long-term dependencies and enhance robust long-horizon forecasting [9].

Multivariate dependencies have also received attention: `CMMamba` adds bidirectional channel-mixing to better integrate inter-variate relationships and enhance feature representation [14]. Sequential Order-Robust Mamba (`SOR-Mamba`) mitigates sequential order bias in channel-wise processing through regularization (minimizing discrepancy from reversed channel orders) and eliminates unnecessary 1D-convolutions [15]. Probabilistic and zero-shot extensions include Mamba-ProbTSF, a dual-network framework for point forecasts plus uncertainty quantification via variance modeling [16], and Mamba4Cast as a foundational backbone trained on synthetic data for efficient zero-shot prediction without task-specific fine-tuning [17].

Emerging applications in power systems further highlight SSM's potential. `PowerMamba` combines deep SSMs with multivariate prediction tailored to electric power dynamics, including renewable integration challenges [18]. Specialized models like Dual-path frequency Mamba-Transformer (`DPFMformer`) target wind power forecasting by capturing rapid fluctuations and meteorological correlations [10], and Wind-Mambaformer leverages hybrid Transformer-Mamba designs for ultra-short-term wind turbine output prediction [7].

Collectively, these developments demonstrate the maturity of Mamba-based models for time series forecasting, providing a strong foundation for domain-specific applications such as wind generation curtailment forecasting, which involves volatile patterns, long memory effects, and multivariate interactions.

3. Dataset description

The analyses presented in this work utilize a specialized open dataset published by the ONS, entitled *Dados de Restrição de Operação por Constrained-off de Usinas Eólicas* (Operational Restriction Data Due to Constrained-off of Wind Power Plants) [11]. All data used in this study were retrieved from the official ONS open data portal at: https://dados.ons.org.br/dataset/restricao_coff_eolica_usi. The analyses reported in the following sections are based on the complete set of monthly files available up to February 2026.

The Brazilian power system is dispatched on an hourly basis, making this forecasting horizon appropriate for analyzing generation constraints. To demonstrate that our model meets the forecasting needs for the horizon used in power planning, we considered a horizon of 1 to 5 steps ahead, corresponding to up to 5 h ahead-of-time forecasting.

3.1. Signal decomposition via EG-VMD

Forecasting wind-curtailment time series is complicated by the superposition of multiple oscillatory phenomena operating at distinct time scales, diurnal wind patterns, weekly demand cycles, seasonal transmission loading, and irregular grid events. To disentangle these components prior to model training, we employ the Entropy-Guided Variational Mode Decomposition (EG-VMD) framework, whose complete pipeline is depicted in Fig. 1.

Background Classical VMD [19] decomposes a real-valued signal $x(t)$ into K band-limited Intrinsic Mode Functions (IMFs) $\{u_k(t)\}_{k=1}^{K}$ by solving the constrained optimization problem

$$\min_{\{u_k\},\{\omega_k\}} \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \quad \text{s.t.} \quad \sum_{k=1}^{K} u_k = x, \tag{1}$$

where ω_k denotes the center frequency of mode k and $\alpha > 0$ is a bandwidth penalty (quadratic regularization weight). The augmented Lagrangian formulation is solved iteratively via the Alternating Direction Method of Multipliers (ADMM), updating mode spectra and center frequencies in the Fourier domain.

Entropy-based hyperparameter selection Two critical hyperparameters, the number of modes K and the penalty α, are conventionally set by the user or tuned via grid search with heuristic criteria. We replace this with an information-theoretic approach (see Fig. 1). Given the input signal x, we first compute a normalized complexity index

$$\mathcal{H} = w_{\mathrm{PE}}\, H_{\mathrm{PE}}(x) \;+\; w_{\mathrm{DE}}\, H_{\mathrm{DE}}(x), \tag{2}$$

where H_{PE} is the permutation entropy [20] of embedding dimension m and delay d, and H_{DE} is the dispersion entropy [21] with c symbol classes. The complexity index $\mathcal{H} \in [0, 1]$ is mapped linearly to an initial penalty estimate $\alpha_0 = \alpha_{\min} + \mathcal{H}(\alpha_{\max} - \alpha_{\min})$: higher signal complexity warrants a larger α to enforce sharper spectral separation among modes.

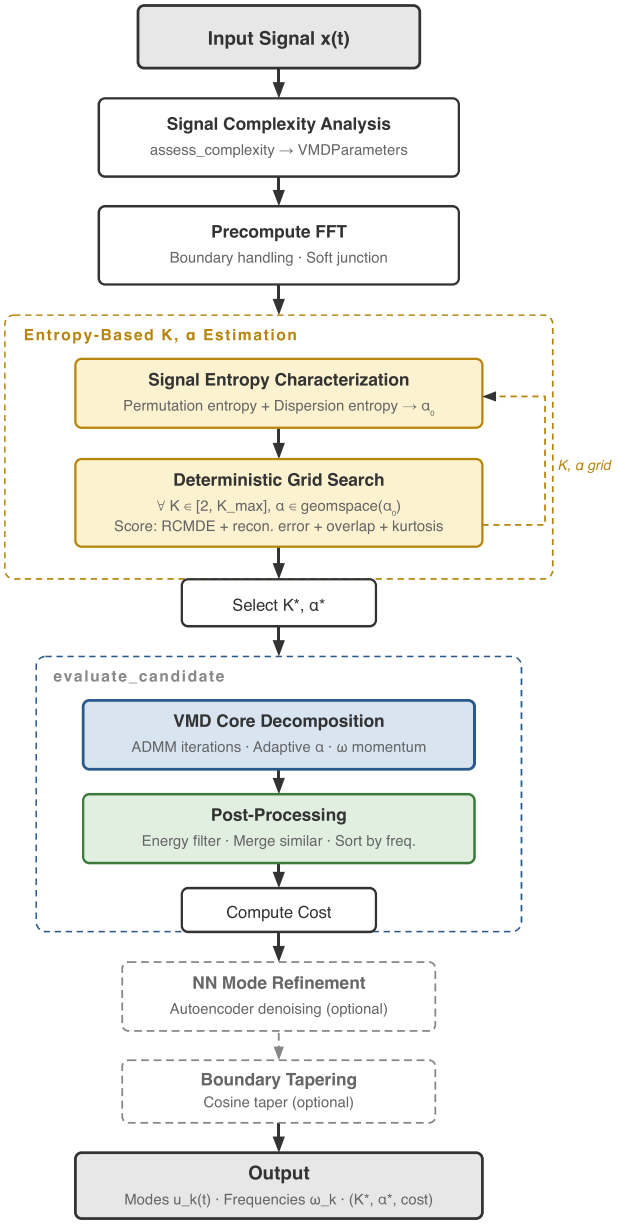


Fig. 1. Entropy-based VMD optimization pipeline. The signal complexity is first characterized via permutation and dispersion entropy, yielding an initial bandwidth penalty α_0. A deterministic grid search over the number of modes K and penalty parameter α selects the configuration that minimizes a composite score based on refined composite multiscale dispersion entropy (RCMDE), reconstruction error, frequency-overlap penalty, and kurtosis reward. The selected (K^*, α^*) are passed to the VMD core, which performs ADMM-based decomposition with adaptive α and frequency-center momentum. Post-processing merges spectrally similar modes, filters negligible-energy components, and optionally applies autoencoder-based denoising.

A deterministic grid search then evaluates every pair (K, α) with $K \in \{2, \dots, K_{\max}\}$ and α drawn from a geometrically spaced grid centered on α_0. Each candidate is scored by a composite objective

$$S(K,\alpha) = \underbrace{0.55\,\overline{E}_{\mathrm{RCMDE}}}_{\text{mode regularity}} + \underbrace{0.18\,\epsilon_{\mathrm{rec}}}_{\text{recon. error}} + \underbrace{0.12\,(1-E_{\mathrm{res}})}_{\text{residual randomness}} + \underbrace{0.08\,\mathcal{O}_{\mathrm{freq}}}_{\text{overlap penalty}} + \underbrace{0.07\,\frac{K}{K_{\max}}}_{\text{parsimony}} - \underbrace{0.20\,\mathcal{K}_w}_{\text{kurtosis reward}}, \tag{3}$$

where $\overline{E}_{\mathrm{RCMDE}}$ is the average refined composite multiscale dispersion entropy across modes (lower values indicate more structured, well-separated IMFs), $\epsilon_{\mathrm{rec}} = \|x - \sum_k u_k\|^2 / \|x\|^2$ is the normalized reconstruction error, E_{res} is the RCMDE of the reconstruction residual (high values are desirable, indicating that only noise remains), $\mathcal{O}_{\mathrm{freq}}$ penalizes excessive spectral overlap between adjacent modes, and $\mathcal{K}_w$ is a frequency-weighted kurtosis reward that favors impulsive content in higher-frequency modes. The pair $(K^*, \alpha^*) = \arg\min S$ is selected for the final decomposition.

The weights in (3) encode a deliberate priority ordering. Mode regularity ($w = 0.55$) is assigned the largest weight because well-separated, structured IMFs are the primary objective of VMD; a decomposition with high reconstruction fidelity but poorly defined modes is of limited analytical value. The kurtosis reward ($w = 0.20$) is the second largest contributor because impulsive, physically interpretable content in higher-frequency modes is the dominant quality indicator in the energy-system signals considered here. Reconstruction fidelity ($w = 0.18$) ranks third to guarantee signal integrity, while residual randomness ($w = 0.12$), spectral overlap ($w = 0.08$), and parsimony ($w = 0.07$) act as regularizers that prevent degenerate solutions without overriding the primary decomposition-quality signal.

Post-processing and mode roles After decomposition, modes whose energy falls below a floor $\epsilon_{\mathrm{floor}}$ relative to the input signal energy are discarded. Remaining modes with dominant frequencies closer than a tolerance $\Delta f_{\mathrm{merge}}$ are merged by summation, and the final set is sorted in ascending frequency order [22]. An optional lightweight autoencoder (single hidden layer) can be applied per-mode for residual denoising, followed by cosine boundary tapering to suppress edge artifacts introduced by the finite-length fast Fourier transform.

4. Methodology

We introduce the *Unified Mamba Forecaster* (UMF), a versatile and modular framework for multivariate time series forecasting, leveraging the selective state-space model Mamba [8] as its core sequence modeling component. Given an input tensor $\mathbf{X} \in \mathbb{R}^{B\times L\times D}$, where B denotes the batch size, L the input sequence length (context window), and D the number of variates, UMF generates a forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{B\times H\times D}$ over a prediction horizon H. The architecture comprises five configurable modules: normalization, decomposition, tokenization, sequence modeling, and projection, each of which can be independently enabled or customized to adapt to diverse forecasting scenarios. A schematic illustration of the architecture is provided in Fig. 2.

4.1. Reversible instance normalization

A prevalent challenge in time series forecasting is the distribution shift between training and inference windows [23]. To mitigate this, we employ Reversible Instance Normalization (RevIN), which performs per-variate normalization along the temporal dimension for each sample. For the dth variate of the bth sample, the normalization proceeds as follows:

$$\mu_{b,d} = \frac{1}{L}\sum_{t=1}^{L} x_{b,t,d}, \tag{4}$$

$$\sigma_{b,d} = \sqrt{\frac{1}{L}\sum_{t=1}^{L}(x_{b,t,d} - \mu_{b,d})^2 + \epsilon}, \tag{5}$$

$$\tilde{x}_{b,t,d} = \gamma_d \cdot \frac{x_{b,t,d} - \mu_{b,d}}{\sigma_{b,d}} + \beta_d, \tag{6}$$

where $\gamma_d, \beta_d \in \mathbb{R}$ are learnable affine parameters, initialized to $\gamma_d = 1$ and $\beta_d = 0$ so that the transformation reduces to standard instance normalization at the start of training, and $\epsilon > 0$ is a small constant for numerical stability (typically $\epsilon = 10^{-5}$). The computed statistics $(\mu_{b,d}, \sigma_{b,d})$ are preserved for the inverse transformation applied to the normalized

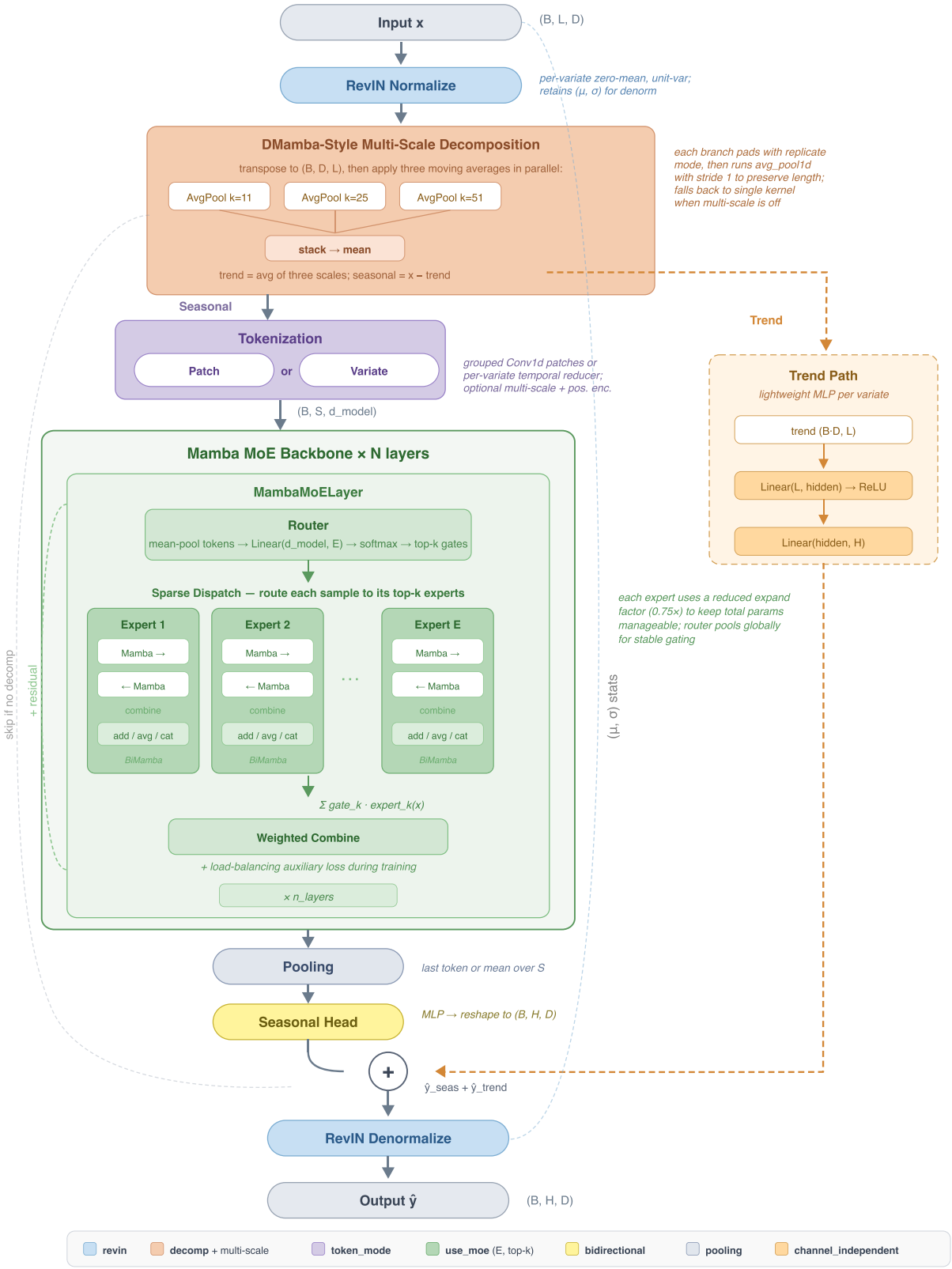


Fig. 2. Overview of the Unified Mamba Forecaster architecture, highlighting the modular components and data flow. Here, S denotes the number of tokens after tokenization, and d_{model} denotes the model embedding dimension used throughout the backbone.

forecast $\hat{\mathbf{Y}}^{\text{norm}}$:

$$\hat{y}_{b,t,d} = \frac{\hat{y}^{\text{norm}}_{b,t,d} - \beta_d}{\gamma_d + \epsilon} \cdot \sigma_{b,d} + \mu_{b,d}. \tag{7}$$

This reversible mechanism ensures that internal representations remain approximately stationary (mean-zero, unit-variance) during the forward pass, enhancing model stability and generalization, while exactly recovering the original scale at output, without information loss.

4.2. Multi-scale series decomposition

Drawing from decomposition-based forecasting paradigms [24] and multi-resolution trend extraction methods [13], we incorporate an optional multi-scale decomposition module. This separates the normalized input $\tilde{\mathbf{X}} \in \mathbb{R}^{B\times L\times D}$ into a trend component $\mathbf{T} \in \mathbb{R}^{B\times L\times D}$, capturing low-frequency variations, and a seasonal residual $\mathbf{S} \in \mathbb{R}^{B\times L\times D}$, encapsulating high-frequency fluctuations:

$$T^{(m)}_{b,t,d} = \frac{1}{k_m} \sum_{j=-\lfloor k_m/2 \rfloor}^{\lfloor k_m/2 \rfloor} \tilde{x}_{b,t+j,d}, \quad m = 1, \dots, M, \tag{8}$$

$$T_{b,t,d} = \frac{1}{M} \sum_{m=1}^{M} T^{(m)}_{b,t,d}, \tag{9}$$

$$S_{b,t,d} = \tilde{x}_{b,t,d} - T_{b,t,d}, \tag{10}$$

where $\{k_m\}_{m=1}^{M}$ are distinct odd kernel sizes for M symmetric moving-average filters, and boundary effects are mitigated via replicate or reflect padding.

This ensemble approach yields a robust trend estimate by averaging across scales, reducing sensitivity to kernel selection. In our default configuration, $M = 3$ with $k_m \in \{11, 25, 51\}$, balancing short-, medium-, and long-term trends. Disabling multi-scale reverts to single-kernel decomposition; fully omitting decomposition sets $\mathbf{S} = \tilde{\mathbf{X}}$ and bypasses the trend pathway. The seasonal $\mathbf{S}$ proceeds to tokenization and backbone processing, while $\mathbf{T}$ is routed to a specialized lightweight head (Section 4.5), allowing the model to allocate capacity efficiently between predictable trends and complex seasonality.

4.3. Tokenization

The tokenization module embeds the seasonal component $\mathbf{S} \in \mathbb{R}^{B\times L\times D}$ into a sequence $\mathbf{Z}^{(0)} \in \mathbb{R}^{B\times S\times d_{\text{model}}}$ of S tokens in model dimension d_{model}. Two strategies are supported, enabling flexibility in handling temporal versus cross-variate dependencies.

4.3.1. Patch mode

Adopting a channel-independent patching strategy [25], each variate is segmented into overlapping temporal patches via a grouped 1D convolution:

$$\mathbf{P} = \text{GroupConv1d}(\mathbf{S}^{\top}; \text{in} = D, \text{out} = d_{\text{model}}, \text{groups} = D, k = p, s = s) \in \mathbb{R}^{B\times d_{\text{model}}\times N_p}, \tag{11}$$

where $N_p = \lfloor (L - p)/s \rfloor + 1$ is the number of patches per variate, and d_{model} must be divisible by D to allocate d_{model}/D channels per group. For multi-resolution patching, M pairs $\{(p_i, s_i)\}_{i=1}^{M}$ generate separate patch sequences, merged via concatenation or interleaving to yield $S = \sum_i N_p^{(i)}$. To encode positional information, additive embeddings $\mathbf{E} \in \mathbb{R}^{1\times S\times d_{\text{model}}}$ are incorporated, either as learnable parameters or fixed sinusoidal functions [26]:

$$E_{n,2k} = \sin\left(n/10000^{2k/d_{\text{model}}}\right), \quad E_{n,2k+1} = \cos\left(n/10000^{2k/d_{\text{model}}}\right), \tag{12}$$

for position $n = 1, \dots, S$ and dimension index k. These may be omitted if the backbone exhibits order invariance.

4.3.2. Variate mode

To prioritize cross-variate modeling, each variate is compressed into a single token ($S = D$), with temporal reduction via optional average pooling (kernel $q > 1$, yielding $L' = \lceil L/q \rceil$) followed by a per-variate reducer. The convolutional variant applies:

$$\mathbf{h}_d = \text{AdaptiveAvgPool1d}\left(\text{GELU}\left(\text{Conv1d}(S_{b,d,:}; k = 3, \text{padding} = 1)\right), 1\right) \in \mathbb{R}^{d_{\text{model}}}, \tag{13}$$

while the MLP variant computes:

$$\mathbf{h}_d = \mathbf{W}_2\, \text{GELU}(\mathbf{W}_1 S_{b,d,:} + \mathbf{b}_1) + \mathbf{b}_2, \tag{14}$$

with $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{hidden}}\times L'}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{model}}\times d_{\text{hidden}}}$. This mode facilitates efficient modeling of inter-variate correlations for high-dimensional series.

4.4. Mixture-of-experts Mamba backbone

The tokenized sequence $\mathbf{Z}^{(0)}$ is transformed through N stacked layers, each comprising RMS normalization [27] and a Mamba block (or MoE extension), with residual connections:

$$\mathbf{Z}^{(\ell)} = \mathbf{Z}^{(\ell-1)} + \text{Mamba}\left(\text{RMSNorm}(\mathbf{Z}^{(\ell-1)})\right), \quad \ell = 1, \dots, N. \tag{15}$$

The Mamba layer [8] realizes a selective SSM, where input-dependent parameters govern the discrete-time dynamics:

$$\mathbf{h}_t = \bar{\mathbf{A}}_t \mathbf{h}_{t-1} + \bar{\mathbf{B}}_t \mathbf{u}_t, \tag{16}$$

$$\mathbf{o}_t = \mathbf{C}_t \mathbf{h}_t, \tag{17}$$

with $\mathbf{h}_t \in \mathbb{R}^{d_{\text{state}}\times E_{\text{exp}}}$, $\bar{\mathbf{A}}_t, \bar{\mathbf{B}}_t, \mathbf{C}_t$ derived via selective gating on input $\mathbf{u}_t \in \mathbb{R}^{d_{\text{model}}}$, state dimension d_{state} (typically small, e.g., 16), and expansion factor E_{exp} (usually 2). The recurrence is efficiently parallelized using associative scans for training.

4.4.1. Bidirectional processing

To capture non-causal dependencies, each Mamba instance may be bidirectional: forward and reverse passes are computed separately and fused:

$$\mathbf{o}_{\text{fwd}} = \text{Mamba}(\mathbf{u}), \tag{18}$$

$$\mathbf{o}_{\text{rev}} = \text{flip}\left(\text{Mamba}(\text{flip}(\mathbf{u}))\right), \tag{19}$$

$$\mathbf{o} = f_{\text{combine}}(\mathbf{o}_{\text{fwd}}, \mathbf{o}_{\text{rev}}), \tag{20}$$

where f_{combine} is element-wise addition, averaging, or linear projection of the concatenation (doubling dimension temporarily).

4.4.2. Sparse mixture-of-experts routing

For enhanced capacity, each layer employs E specialized Mamba experts (optionally bidirectional), with a router selecting top-K per sample. The router aggregates tokens via mean pooling and computes softmax logits:

$$\mathbf{g}_b = \text{softmax}\left(\mathbf{W}_r \cdot \left(\frac{1}{S}\sum_{s=1}^{S} \mathbf{Z}^{(\ell-1)}_{b,s,:}\right) + \mathbf{b}_r\right) \in [0, 1]^E, \tag{21}$$

with $\mathbf{W}_r \in \mathbb{R}^{E\times d_{\text{model}}}$, $\mathbf{b}_r \in \mathbb{R}^E$. Top-$K$ indices $\mathcal{T}_b = \text{top}{-}\text{K}(\mathbf{g}_b)$ are selected, gates renormalized:

$$\hat{g}_{b,e} = \frac{g_{b,e}}{\sum_{e'\in\mathcal{T}_b} g_{b,e'}}, \quad \forall e \in \mathcal{T}_b, \tag{22}$$

and outputs combined residually:

$$\mathbf{Z}^{(\ell)}_b = \mathbf{Z}^{(\ell-1)}_b + \sum_{e\in\mathcal{T}_b} \hat{g}_{b,e} \cdot \text{Expert}_e\left(\mathbf{Z}^{(\ell-1)}_b\right). \tag{23}$$

Experts use reduced expansion αE_{exp} ($\alpha < 1$, e.g., 0.75) for parameter efficiency. To promote balanced utilization, an auxiliary loss penalizes imbalance:

$$\mathcal{L}^{(\ell)}_{\text{aux}} = \lambda_{\text{aux}} \cdot E \cdot \sum_{e=1}^{E} f_e \cdot \bar{g}_e, \tag{24}$$

where $f_e = |\{b : e \in \mathcal{T}_b\}|/B$, $\bar{g}_e = B^{-1}\sum_b g_{b,e}$, and $\lambda_{\text{aux}} \approx 0.01$. This quadratic form is minimized under uniform load ($f_e = \bar{g}_e = 1/E$). Optionally, a post-SSM channel mixer enhances feature interactions:

$$\mathbf{Z}^{(\ell)} \leftarrow \mathbf{Z}^{(\ell)} + \text{FFN}\left(\text{RMSNorm}(\mathbf{Z}^{(\ell)})\right), \tag{25}$$

with $\text{FFN}(\mathbf{z}) = \mathbf{W}_2\,\text{GELU}(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2$, hidden dimension $r d_{\text{model}}$.

4.5. Pooling and prediction heads

The backbone output $\mathbf{H} = \mathbf{Z}^{(N)} \in \mathbb{R}^{B\times S\times d_{\text{model}}}$ is aggregated to a context vector $\mathbf{c}_b \in \mathbb{R}^{d_{\text{model}}}$ via last-token extraction or mean pooling:

$$\mathbf{c}_b = \begin{cases} \mathbf{H}_{b,S,:} & \text{(last)}, \\ S^{-1}\sum_{s=1}^{S}\mathbf{H}_{b,s,:} & \text{(mean)}. \end{cases} \tag{26}$$

The seasonal forecast is obtained via an MLP head:

$$\hat{\mathbf{Y}}_b^{\text{seas}} = \text{reshape}\left(\text{MLP}(\mathbf{c}_b), H \times D\right) \in \mathbb{R}^{H\times D}, \tag{27}$$

where the final layer maps to $H\,D$ dimensions. If decomposition is active, the trend is predicted per variate using a two-layer MLP:

$$\hat{\mathbf{Y}}_{b,:,d}^{\text{trend}} = \mathbf{W}_4\,\text{ReLU}(\mathbf{W}_3\mathbf{T}_{b,:,d} + \mathbf{b}_3) + \mathbf{b}_4 \in \mathbb{R}^{H}, \tag{28}$$

with $\mathbf{W}_3 \in \mathbb{R}^{h_t\times L}$, $\mathbf{W}_4 \in \mathbb{R}^{H\times h_t}$, h_t small (e.g., 64). The composite forecast is denormalized:

$$\hat{\mathbf{Y}}_b = \text{RevIN}^{-1}\left(\hat{\mathbf{Y}}_b^{\text{seas}} + \mathbb{1}_{\text{decomp}}\hat{\mathbf{Y}}_b^{\text{trend}}; \mu_b, \sigma_b\right), \tag{29}$$

where $\mathbb{1}_{\text{decomp}} = 1$ if decomposition is enabled, else 0.

5. Results

We present results in three parts: an ablation study isolating the contribution of each architectural component Section 5.2, benchmark comparisons against established baselines Section 5.3, and diagnostic analyses characterizing the model's internal behavior Section 5.4. Before proceeding, we describe the preprocessing pipeline applied to the raw time series.

The performance of the evaluated models was assessed using MSE, Root Mean Squared Error (RMSE), and MAE. These metrics were selected because they provide complementary perspectives on forecasting performance, with MSE emphasizing larger errors due to the squared term, RMSE preserving the original scale of the data while maintaining sensitivity to large deviations, and MAE offering an interpretable measure of the average absolute prediction error [28].

5.1. Preprocessing

Real-world time series often exhibit non-stationarity, heterogeneous scales, and a mixture of low-frequency trends with higher-frequency oscillatory components. To disentangle these, we apply VMD to extract a set of IMFs that partition the signal's spectral content. Fig. 3 illustrates this pipeline on a representative sequence. The top panel shows the original time series alongside the extracted trend, obtained as the lowest-frequency IMF (mode 0). The bottom panel displays the detrended signal, computed by subtracting the trend from the original, alongside a filtered version that retains only the dominant IMFs. This decomposition serves two purposes: it removes slow-varying drift that can dominate the loss landscape and obscure seasonal patterns, and it provides a cleaner target for the model to learn. The filtered, detrended series is used as input to all models in our experiments.

5.2. Ablation study

We conduct a systematic ablation over the four main architectural toggles, tokenization mode (patch vs. variate), channel mixer, series decomposition, and RevIN, evaluating all $2^4 = 16$ combinations under identical training conditions with a single Mamba backbone. Results are reported in Table 1, sorted by test MSE.

The most striking pattern is the dominant effect of RevIN. Every configuration with RevIN enabled occupies the top half of the ranking, while every configuration without it falls into the bottom half. The best RevIN-off variant (MSE = 0.5695) underperforms the worst RevIN-on variant (0.5047) by a margin exceeding 0.06, confirming prior findings [23] that per-variate instance normalization is necessary for stable optimization when input variates exhibit heterogeneous scales and non-stationary statistics. All subsequent analysis conditions on RevIN being enabled.

Within the RevIN-on regime, decomposition emerges as the second most impactful component. Comparing matched configuration pairs

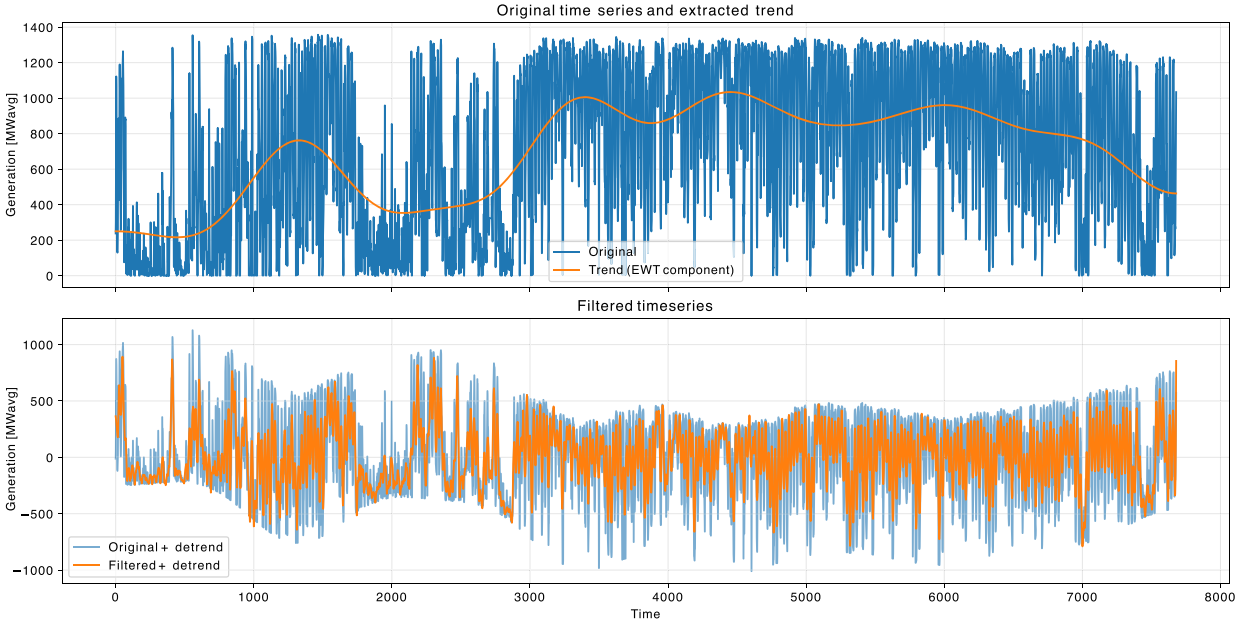


Fig. 3. Wind curtailment time series preprocessing results via VMD. **Top:** Original time series (blue) and extracted trend component corresponding to the lowest-frequency IMF (orange). The trend captures slow-varying drift over thousands of time steps. **Bottom:** Detrended signal (blue) obtained by subtracting the trend, and filtered signal (orange) reconstructed from the sum of dominant IMFs. Filtering attenuates high-frequency noise while preserving the primary oscillatory structure that the model is trained to forecast.

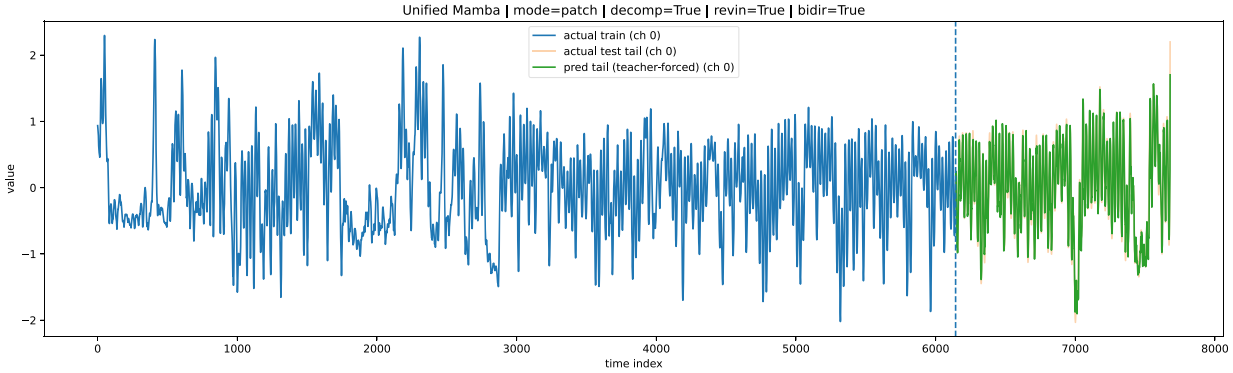


Fig. 4. Wind curtailment forecasting prediction on a representative test sequence (normalized data). Blue shows the training portion of the time series (channel 0); orange indicates the held-out test segment; green shows the model's predictions under teacher forcing. The vertical dashed line marks the train/test boundary. The model accurately tracks both the amplitude and phase of oscillations in the test region, with no visible drift or systematic bias.

Table 1
Ablation study on architectural components. Results sorted by Test MSE.

Token Mode	Channel Mixer	Decomp	RevIN	Test MAE ↓	Test MSE ↓
Patch	✓	✓	✓	**0.5358**	**0.4618**
Variate	✗	✓	✓	0.5467	0.4799
Variate	✓	✓	✓	0.5531	0.4871
Patch	✗	✗	✓	0.5461	0.4873
Variate	✗	✗	✓	0.5486	0.4885
Variate	✓	✗	✓	0.5524	0.4969
Patch	✓	✗	✓	0.5649	0.5014
Patch	✗	✓	✓	0.5612	0.5047
Variate	✗	✓	✗	0.5980	0.5695
Patch	✗	✗	✗	0.5977	0.5813
Patch	✓	✗	✗	0.6068	0.5876
Patch	✓	✓	✗	0.6019	0.5878
Variate	✗	✗	✗	0.6064	0.5918
Variate	✓	✗	✗	0.6034	0.5959
Variate	✓	✓	✗	0.6178	0.6141
Patch	✗	✓	✗	0.6149	0.6282

Normalized data.

differing only in decomposition status, the largest improvement appears in patch mode with channel mixer (0.4618 vs. 0.5014, a relative gain of 7.9%), with smaller but consistent benefits observed across other configurations (0.4799 vs. 0.4885 in variate mode without mixer). These improvements support the hypothesis that explicitly separating trend and seasonal dynamics allows each prediction head to specialize: the lightweight trend MLP handles slowly varying structure while the Mamba backbone devotes capacity to modeling higher-frequency seasonal patterns. The remainder of this section therefore conditions on patch tokenization with series decomposition, channel mixer, and RevIN enabled, the configuration that achieves the lowest test MSE of 0.4618, and investigates the remaining design dimensions under these fixed settings.

5.3. Benchmark comparison

We evaluate two variants of the proposed architecture against a comprehensive set of established baselines: **MoE Mamba**: Mamba backbone augmented with Mixture-of-Experts routing, enabling conditional computation and expert specialization while preserving linear-time complexity. The routing layer uses $E = 4$ experts with a top-$K = 2$ selection per token, so each input activates exactly half of the expert pool while keeping the remaining experts inactive. **Single Mamba**: Standard Mamba backbone without MoE, included to isolate the contribution of expert routing from other architectural choices.

Table 2 reports MSE, RMSE, and MAE for all models. MoE Mamba achieves state-of-the-art performance, outperforming the strongest prior methods, NBEATSx and NBEATS, by 27% in MSE. The gap widens against other competitive baselines: 45% over NHITS and 50% over PatchTST. Compared to the Single Mamba ablation, the MoE variant improves MSE by 15.6%, MAE by 5.2%, and RMSE by 8.1%, demonstrating that expert specialization provides meaningful gains beyond the base architecture.

Table 2
Results on the evaluated forecasting task. Lower is better for all metrics. Best result in **bold**.

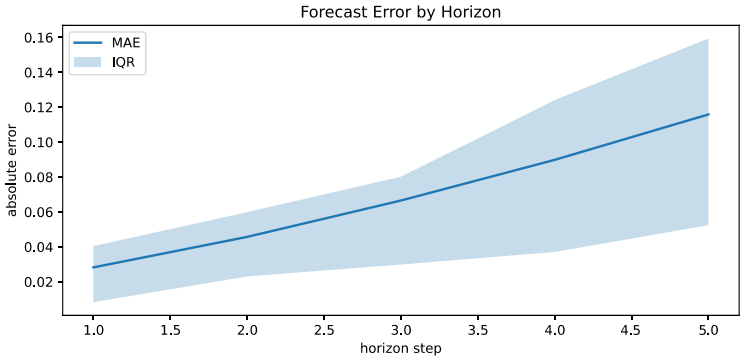
Model	MSE	RMSE	MAE
MoE Mamba (Ours)	**0.0140**	**0.1184**	**0.0805**
Single Mamba (Ours)	0.0166	0.1289	0.0850
NBEATSx	0.0192	0.1386	0.0960
NBEATS	0.0192	0.1386	0.0960
NHITS	0.0254	0.1595	0.1129
PatchTST	0.0279	0.1670	0.1153
TFT	0.1437	0.3791	0.2869
TimesNet	0.1932	0.4395	0.3320
TimeXer	0.2076	0.4556	0.3647
Autoformer	0.2546	0.5046	0.3840
GRU	0.2636	0.5134	0.4316
TimeMixer	0.2755	0.5249	0.4320
iTransformer	0.2964	0.5444	0.4316
TCN	0.3234	0.5687	0.4556
TiDE	0.3299	0.5744	0.4544
LSTM	0.3538	0.5948	0.4864
Transformer	0.3808	0.6171	0.5017
StemGNN	0.3882	0.6231	0.5039
FEDformer	0.3951	0.6286	0.4934
DeepAR	0.4074	0.6383	0.5278
BiTCN	0.4165	0.6454	0.5324
Informer	0.5286	0.7270	0.5921

Normalized data. The baseline models are from the `Nixtla` repository [29].

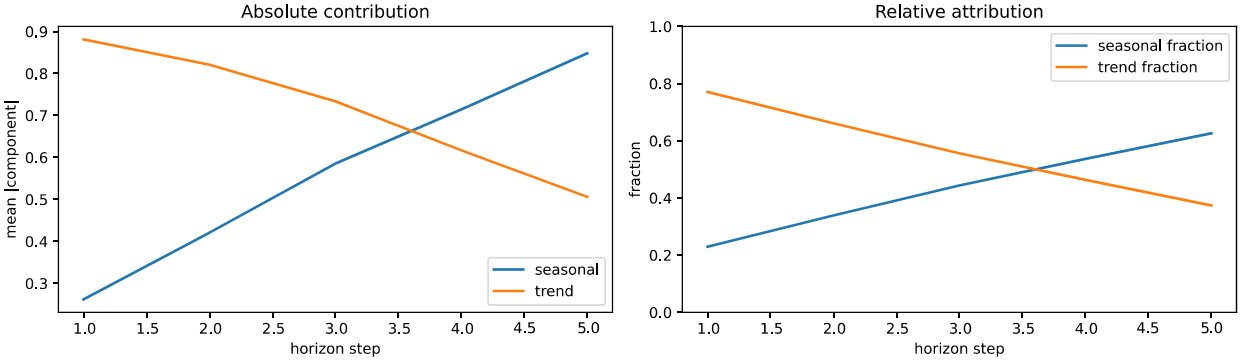
The baseline models used in this study were obtained from the `Nixtla` repository [29], which provides standardized implementations of state-of-the-art time series forecasting methods. To ensure a fair and reproducible comparison, the default hyperparameter configurations provided in the repository were adopted for all baseline models, following the recommended hyperparameter settings defined by the authors.

5.4. Diagnostic analysis

Beyond aggregate metrics, we conduct a series of diagnostic experiments to characterize the internal behavior of the trained model. These analyses address three questions: how does forecast quality degrade across the prediction horizon, what roles do the architectural



(a) Forecasting error by horizon.



(b) Absolute and relative attributions of seasonal and trend params by horizon.

Fig. 5. Horizon-dependent forecast characteristics. **(a)** MAE increases monotonically from ~0.025 at step 1 to ~0.12 at step 5, with a widening interquartile range reflecting growing uncertainty. The approximately linear trajectory indicates stable autoregressive decoding without exponential error accumulation. **(b)** Trend and seasonal attribution (left: absolute magnitude; right: relative fraction). Trend dominates at short horizons (~70% at step 1) while seasonal contribution increases with horizon length, crossing 50% near step 3.5, consistent with trend extrapolation becoming unreliable at longer horizons while periodic structure remains phase-coherent.

components play, and how does the MoE routing mechanism behave in practice?

Fig. 4 provides a qualitative overview of model behavior on a representative test sequence. The predicted trajectory (green) closely tracks the ground truth (orange) throughout the test region, capturing both the amplitude and phase of the underlying oscillations. No systematic drift or bias is visible, and the model maintains fidelity even as the forecast extends over several hundred time steps. This visual inspection complements the quantitative analyses that follow.

5.4.1. *Forecast error and component attribution*

Fig. 5 characterizes how forecast quality and component contributions evolve across the prediction horizon. Panel (a) displays MAE as a function of horizon step, with shaded bands indicating the interquartile range across test samples. Error increases monotonically from approximately 0.025 at horizon step 1 to 0.12 at horizon step 5, accompanied by a widening IQR that reflects growing predictive uncertainty. The approximately linear error trajectory indicates that the autoregressive decoding procedure does not suffer from the exponential error accumulation that can arise in recurrent forecasters when small prediction errors compound across steps.

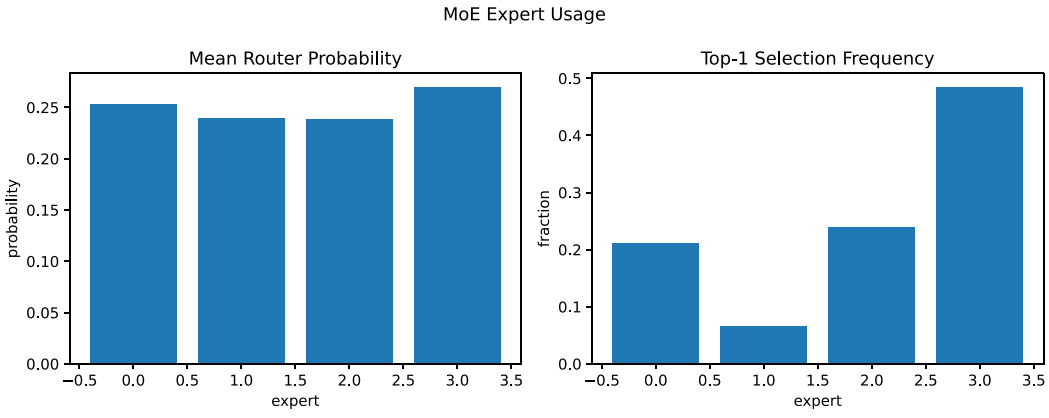
Panel (b) decomposes predictions into contributions from the trend and seasonal heads, reporting both absolute magnitude (left) and relative fraction (right). At the first horizon step, the trend component accounts for approximately 70% of output magnitude while the seasonal component contributes 30%. These proportions reverse as the horizon extends: the seasonal fraction crosses 50% near step 3.5 and reaches approximately 65% by step 5. This crossover aligns with the design rationale for dual-head decomposition. Trend extrapolation relies on continuing a locally-estimated slope, which becomes increasingly unreliable as the forecast extends beyond the context window. Seasonal structure, by contrast, is periodic and phase-coherent: once the model learns the relevant frequencies, it can project them forward with less degradation.

5.4.2. *Expert utilization and routing dynamics*

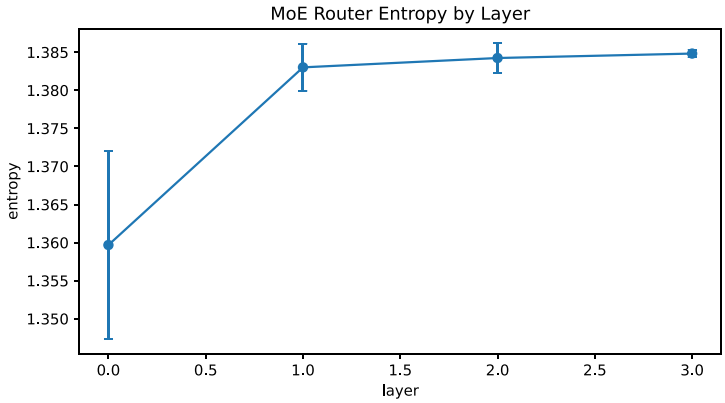
Fig. 6 characterizes expert utilization and routing entropy across the MoE layers. Panel (a) shows that mean router probabilities are approximately uniform at ~0.25 per expert, indicating that the gating network does not collapse to a degenerate solution at the soft-assignment level. However, top-1 selection frequencies reveal a different picture: expert 3 accounts for approximately 48% of all hard routing decisions while expert 1 is selected for only 7%. This discrepancy between soft and hard routing reflects the winner-take-all property of sparse mixture models, where the argmax operation amplifies small logit differences into concentrated selections.

Panel (b) examines how routing behavior varies with network depth. Router entropy increases monotonically from layer 0 ($H \approx 1.36$) to layer 3 ($H \approx 1.385$), approaching the maximum entropy for four experts ($\log 4 \approx 1.386$). This gradient indicates that early layers route more deterministically, concentrating computation on a smaller subset of experts, while deeper layers distribute probability mass more uniformly. A plausible interpretation is that early layers perform coarse-grained specialization based on broad input characteristics, while later layers engage more diverse expert combinations to capture fine-grained patterns.

To assess routing robustness, we perturb input representations by adding Gaussian noise with standard deviation equal to 1% of the representation norm. Across eight independent trials, the router maintains 100% top-1 consistency, with KL divergence between clean and perturbed distributions on the order of 10^{-7}. This stability indicates that



(a) Expert utilization



(b) Router entropy by layer

Fig. 6. MoE routing behavior. **(a)** Expert utilization aggregated across layers. Left: mean router probability per expert (∼0.25 each), indicating balanced soft assignment. Right: top-1 selection frequency, revealing concentration on expert 3 (48%) despite uniform soft routing, reflecting the winner-take-all property of sparse mixtures. **(b)** Router entropy increases from $H \approx 1.36$ at layer 0 to $H \approx 1.385$ at layer 3 (maximum $\log 4 \approx 1.386$), indicating that early layers route deterministically while deeper layers distribute probability more uniformly.

the learned routing function operates far from decision boundaries and is unlikely to exhibit erratic behavior under minor distributional shifts.

6. Conclusion

This work introduced a decomposition-driven, expert-routed Mamba framework for wind-constrained forecasting using real operational data from the National Interconnected System of Brazil provide by ONS. By integrating entropy-guided VMD with a modular Mamba backbone and sparse MoE routing, the proposed Unified Mamba Forecaster effectively addresses the volatile, multi-scale, and long-memory characteristics of wind curtailment time series.

The empirical results demonstrate three principal findings. First, reversible instance normalization is essential for stable optimization under heterogeneous and non-stationary input distributions. Second, explicit separation of trend and seasonal dynamics significantly enhances predictive performance, particularly when combined with patch-based tokenization and channel interaction modeling. Third, sparse expert routing provides substantial additional gains over a single-backbone Mamba, yielding state-of-the-art performance and outperforming strong baselines such as RNNs, MLPs, and Transformer-based forecasters by large margins in MSE, RMSE, and MAE.

Future work will extend this framework along three directions: probabilistic forecasting with calibrated uncertainty quantification, integration of exogenous meteorological and transmission variables, and transfer learning across regional subsystems to improve robustness under evolving grid conditions. Additionally, investigating adaptive expert allocation and dynamic decomposition strategies may further enhance generalization in rapidly changing renewable penetration scenarios.

The integration of exogenous variables as auxiliary inputs to the MoE Mamba framework will enable the model to better capture the operational conditions that lead to curtailment events. Furthermore, the inclusion of these variables may allow the model to learn more robust cross-domain dependencies between generation, demand, and environmental factors, potentially improving generalization and long-horizon prediction performance.

CRediT authorship contribution statement

Laio Oriel Seman: Writing – original draft, Software, Investigation, Data curation, Conceptualization; **Stefano Frizzo Stefenon:** Writing – review & editing, Supervision; **João Pedro Matos-Carvalho:** Writing – review & editing.

Data availability

The dataset is available at: https://dados.ons.org.br/dataset/restricao_coff_eolica_usi

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by FCT - Fundação para a Ciência e Tecnologia, I.P. under Grants 2023.15441.TENURE.051/CP00003/CT00029 and the LASIGE Research Unit (ref. UID/00408/2025, DOI: https://doi.org/10.54499/UID/00408/2025), and co-financed by FCT and the PRR - Recovery and Resilience Plan by the European Union under the LASIGE Research Unit (UID/PRR/00408/2025, DOI: https://doi.org/10.54499/UID/PRR/00408/2025).

References

[1] L.S. Aquino, L.O. Seman, V.C. Mariani, L.D.S. Coelho, S.F. Stefenon, G.V. González, Spatiotemporal wind energy forecasting: a comprehensive survey and a deep equilibrium-based case study with StemGNN, IEEE Access 13 (2025) 131461–131482.

[2] C. Cacciuttolo, M. Navarrete, D. Cano, Advances, progress, and future directions of renewable wind energy in Brazil (2000–2025–2050), Appl. Sci. 15 (10) (2025) 5646.

[3] G.T.T. Vieira, W.N. Silva, L.F.N. Lourenço, R.M. Monaro, M.B.C. Salles, C.F.M. Almeida, Characterizing wind and solar curtailment in Brazil: an evidence-based analysis of operational drivers, Electr. Power Syst. Res. 254 (2025) 112672.

[4] Wood Mackenzie, Brazil Forecast to Hit 8Transition, Technical Report, PV Tech, 2025.

[5] R. Ferreira, A.T. Coral, B. Meyer, E.C.S. Franco, F.M. Beltrame, Constrained-off energy events in wind farms on Brazilian market, in: CIGRE Session 48, International Council on Large Electric Systems (CIGRE), Paris, France (e-session), 2020, pp. 1–11.

[6] L. Bird, J. Cochran, X. Wang, Wind and Solar Energy Curtailment: Experience and Practices in the United States, Technical Report NREL/TP-6A20-60983, National Renewable Energy Laboratory (NREL), 2014.

[7] Z. Dong, Y. Zhao, W. Chen, Y. Li, X. Han, G. Zhang, J. Wang, Wind-Mambaformer: ultra-short-term wind turbine power forecasting based on advanced transformer and Mamba models, Energies 18 (5) (2025) 1155.

[8] A. Gu, T. Dao, Mamba: linear-time sequence modeling with selective state spaces, 2312.00752 2 (2024) 1–36.

[9] Z. Wu, Y. Gong, A. Zhang, DTMamba: dual twin Mamba for time series forecasting, 2405.07022 1 (2024) 1–9.

[10] J.-T. Hong, S. Han, J. Yan, Y.-Q. Liu, Dual-path frequency Mamba-transformer model for wind power forecasting, Energy 332 (2025) 137225.

[11] Operador Nacional do Sistema Elétrico (ONS), Dados de restrição de operação por constrained-off de usinas eólicas, 2026, (https://dados.ons.org.br/dataset/restricao_coff_eolica_usi).

[12] Z. Wang, F. Kong, S. Feng, M. Wang, X. Yang, H. Zhao, D. Wang, Y. Zhang, Is Mamba effective for time series forecasting?, 2403.11144 3 (2024) 1–14.

[13] R. Chen, F. Sun, DMamba: decomposition-enhanced Mamba for time series forecasting, 2602.09081 1 (2026) 1–9.

[14] Q. Li, J. Qin, D. Cui, D. Sun, D. Wang, CMMamba: channel mixing Mamba for time series forecasting, J. Big Data 11 (2024) 153.

[15] S. Lee, J. Hong, K. Lee, T. Park, Sequential order-robust Mamba for time series forecasting, 2410.23356 1 (2024) 1–12.

[16] P. Pessoa, P. Campitelli, D.P. Shepherd, S.B. Ozkan, S. Pressé, Mamba time series forecasting with uncertainty quantification, Mach. Learn. Sci. Technol. 6 (3) (2025) 35012.

[17] S.K. Bhethanabhotla, O. Swelam, J. Siems, D. Salinas, F. Hutter, Mamba4Cast: efficient zero-shot time series forecasting with state space models, 2410.09385 1 (2024) 1–13.

[18] A. Menati, F. Doudi, D. Kalathil, L. Xie, PowerMamba: a deep state space model and comprehensive benchmark for time series prediction in electric power systems, 2412.06112 2 (2024) 1–14.

[19] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, IEEE Trans. Signal Process. 62 (3) (2014) 531–544.

[20] C. Bandt, B. Pompe, Permutation entropy: a natural complexity measure for time series, Phys. Rev. Lett. 88 (17) (2002) 174102.

[21] M. Rostaghi, H. Azami, Dispersion entropy: a measure for time-series analysis, IEEE Signal Process. Lett. 23 (5) (2016) 610–614.

[22] M.H. D.M. Ribeiro, R.G.d. Silva, S.R. Moreno, C. Canton, J.H.K. Larcher, S.F. Stefenon, V.C. Mariani, L. dos Santos Coelho, Variational mode decomposition and bagging extreme learning machine with multi-objective optimization for wind power forecasting, Appl. Intell. 54 (2024) 3119–3134.

[23] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, J. Choo, Reversible instance normalization for accurate time-series forecasting against distribution shift, in: International Conference on Learning Representations (ICLR), 2022, pp. 1–25.

[24] T. Zhou, Z. Ma, X. Wang, Q. Wen, L. Sun, T. Yao, R. Jin, FEDformer: frequency enhanced decomposed transformer for long-term series forecasting, in: International Conference on Machine Learning (ICML), 162 of *Proceedings of Machine Learning Research*, 2022, pp. 1–19.

[25] Y. Nie, N.H. Nguyen, P. Blache, Y. Kong, A. Pesaranghader, Y. Zhang, J. Zhou, W. Lin, H. Lee, Y. Choi, et al., A time series is worth 64 words: long-term forecasting with transformers, in: International Conference on Learning Representations (ICLR), 2, 2023, pp. 1–24.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), 7, 2017, pp. 1–15.

[27] B. Zhang, R. Sennrich, Root mean square layer normalization, in: Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 12360–12371.

[28] L.O. Seman, S.F. Stefenon, K.-C. Yow, L. dos Santos Coelho, V.C. Mariani, Fourier-enhanced sequence-to-sequence latent graph neural networks for multi-node spatiotemporal forecasting in a hydroelectric reservoir, Eng. Appl. Artif. Intell. 167 (2026) 113939.

[29] K.G. Olivares, C. Challu, F. Garza, M.M. Canseco, A. Dubrawski, NeuralForecast: user friendly state-of-the-art neural forecasting models, 2022, (PyCon Salt Lake City, Utah, USA). https://github.com/Nixtla/neuralforecast.