JACOWAVENET: WAVELET RESIDUAL PYRAMID WITH SIGNED POLYNOMIAL TEMPORAL MAPPING FOR HETEROGENEOUS TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting benefits from modelling long-range dependencies and cross-scale relations, yet multi-scale representations may entangle repeated coarse information with local variation. We propose JacoWaveNet, a dual-subband wavelet residual pyramid that separates approximation and detail components and represents each finer scale by its innovation over a wavelet-consistent coarse lift. On this representation, the Jacobi Mapping (JM) module first performs one signed temporal aggregation and then applies an element-wise Jacobi polynomial basis mapping to its normalized messages. Thus, the polynomial order controls nonlinear basis responses rather than temporal-hop propagation. Experiments on seven benchmarks, together with multi-seed, basis, wavelet, redundancy, scalability, and external-domain analyses, support the value and scope of the proposed design for medium- and long-horizon forecasting.

1 INTRODUCTION

Time series forecasting requires modelling long-range dependencies and cross-scale interactions, particularly when signals exhibit heterogeneous temporal dynamics. In real-world benchmarks such as energy, weather, electricity, and traffic, effective models must simultaneously capture global trends, local fluctuations, and signed temporal relations over medium- and long-term horizons. These patterns are inherently coupled: low-frequency trends may recur across resolutions, while negative learned relation scores can emerge after scale-specific decomposition. When all patterns are encoded into a shared representation, cross-scale feature entanglement can obscure both global and local structures critical for accurate forecasting.

Recent studies have advanced time series forecasting along multiple directions. Transformer-based models enhance long-range dependency modelling through efficient attention mechanisms, temporal patching, periodic variation modelling, inverted-variable representations, and large-model reprogramming (Zhou et al., 2021; Nie et al., 2023; Wu et al., 2023; Liu et al., 2024b; Jin et al., 2024; Wang et al., 2024). Relation-oriented methods explicitly capture temporal or variate interactions via cross-dimension attention, dynamic correlations, or channel permutation-invariant relation modelling (Zhang & Yan, 2023; Cai et al., 2024a; Gao et al., 2024; Xu et al., 2026). Frequency- and wavelet-based approaches further improve cross-scale modelling by separating global and local structures prior to prediction (Liu et al., 2024a; Yang et al., 2023; Huang et al., 2025; Shu & Lampos, 2026). These advances suggest that forecasting accuracy is no longer limited only by the capacity of the sequence backbone. It also depends on whether the input representation exposes the right temporal evidence before the backbone aggregates it. In heterogeneous benchmarks, smooth trends, short-lived fluctuations, and suppressive relations may coexist in the same window but operate at different scales. If they are compressed too early into a single latent state, subsequent attention, mixing, or graph-style aggregation has to recover multiple incompatible structures from an already entangled representation.

Despite these advances, two key limitations remain. First, existing multi-scale representations often retain redundant low-frequency semantics across adjacent scales, leading to persistent feature entanglement rather than structured representations. Second, many temporal aggregation mechanisms do not distinguish positive and negative learned relation scores, causing opposing temporal evidence to be conflated. These two issues interact with each other. Redundant low-frequency content can make

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

adjacent scales appear correlated even when their useful predictive signals differ, while unmodeled negative dependencies can make local reversals look like noise rather than structured temporal evidence. As a result, a model may fit average trends but still miss horizon-specific changes that dominate medium- and long-term prediction.

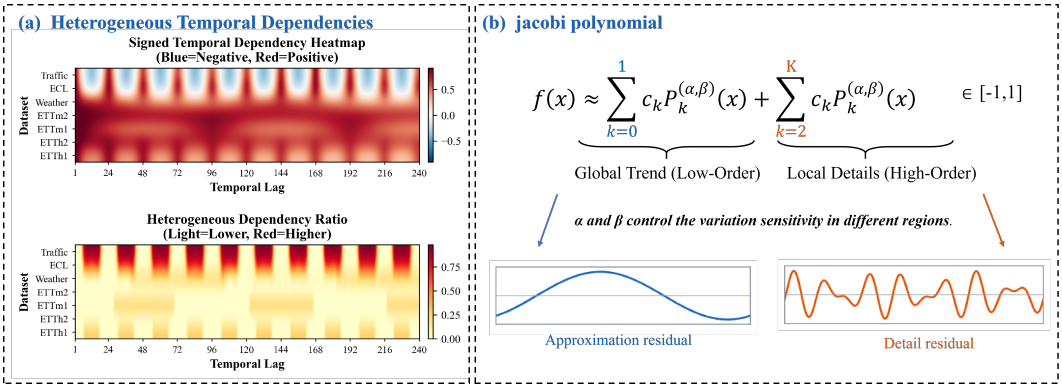


Figure 1: Motivation for heterogeneous temporal relations and Jacobi polynomial mapping. (a) Signed temporal-relation scores across benchmarks, where blue and red denote negative and positive scores over temporal lags, respectively. (b) Jacobi basis responses on a bounded message domain; increasing the basis order increases response complexity rather than the number of temporal hops.

To address these issues, we advocate a representation-first design. We propose JacoWaveNet, a Wavelet Residual Pyramid framework with signed polynomial temporal mapping for heterogeneous time series forecasting. The wavelet component decomposes signals into approximation and detail subbands and constructs residual pyramids for each. This design makes a finer level encode the information not explained by a wavelet-consistent coarse lift while preserving scale-specific dynamics through subband decoupling. The resulting representation provides a structured multi-scale foundation for subsequent temporal modelling. Figure 1 illustrates this motivation from two aspects. Panel (a) shows that temporal-relation scores vary across datasets and lags and may be positive or negative. Panel (b) illustrates the Jacobi basis on the normalized message domain. JacoWaveNet constructs a dual-subband residual representation and uses Jacobi Mapping (JM) to process signed temporal evidence within this disentangled space.

Building upon this representation, JM first routes one signed temporal message and then applies an element-wise Jacobi expansion to its normalized values. The Jacobi order therefore specifies nonlinear basis responses after routing; it does not form a polynomial of the temporal-relation matrix or add temporal hops. The two modules are complementary: the residual pyramid separates scale-specific information before JM forms temporal messages, so the basis mapping need not perform scale decomposition itself. Our ablations, multi-seed results, basis and wavelet sensitivity, and redundancy diagnostics test these distinct roles. Our contributions are summarized as follows:

- We propose JacoWaveNet, a dual-subband wavelet residual pyramid framework that constructs structured multi-scale representations by explicitly separating approximation and detail components.
- We develop a Jacobi Mapping (JM) module that combines one signed temporal aggregation with an element-wise Jacobi basis mapping on normalized messages.
- Experiments on seven benchmarks and an external air-quality dataset include controlled ablations, multi-seed uncertainty, basis and wavelet sensitivity, redundancy diagnostics, and scalability analysis.

2 Related Work

2.1 Cross-Scale and Cross-Variable Learning

Long-horizon forecasting has advanced from recurrent and hierarchical baselines such as LSTNet, DeepAR, and N-HiTS to attention-based and pattern-mixing models (Lai et al., 2018; Salinas et al., 2020; Challu et al., 2023; Vaswani et al., 2017). Informer, PatchTST, TimesNet, and iTransformer improve long-range or multivariate modeling through sparse attention, patching, temporal 2D variation, and inverted variate tokens (Zhou et al., 2021; Nie et al., 2023; Wu et al., 2023; Liu et al., 2024b). Recent pattern-mixing and foundation-style models, including TimeMixer, TimeMixer++, xLSTM-Mixer, Timer, Moirai, MOMENT, and UniTS, further study cross-scale patterns and universal forecasting representations (Wang et al., 2024; 2025; Kraus et al., 2024; Liu et al., 2024c; Woo et al., 2024; Goswami et al., 2024; Gao et al., 2024). Relation-oriented methods such as MTGNN, Crossformer, MSGNet, MSCGrapher, and CPiRi explicitly model cross-dimension or inter-series interactions (Wu et al., 2020; Zhang & Yan, 2023; Cai et al., 2024a; Yang et al., 2025; Xu et al., 2026). These methods validate relation modeling, but many still aggregate heterogeneous temporal evidence in a shared latent space or a single relation map. In contrast, JacoWaveNet first constructs scale-specific residual subbands and then applies signed Jacobi Mapping within this decoupled workspace. Its relation scores are temporal within each variable stream rather than an explicit cross-variable graph.

2.2 Decomposition-Based Forecasting Models

Decomposition is useful because real-world time series contain trends, seasonality, periodicity, and local fluctuations. Autoformer introduces decomposition into Transformers, FEDformer uses frequency-enhanced decomposition, DLinear highlights simple trend-seasonal separation, and TimeMixer performs decomposable multi-scale mixing (Wu et al., 2021; Zhou et al., 2022; Zeng et al., 2023; Wang et al., 2024). Spectral and wavelet-based methods such as WFTNet, WaveForM, SPTGNN, and Sonnet further expose scale-specific structures (Liu et al., 2024a; Yang et al., 2023; Jin et al., 2025; Shu & Lampos, 2026). Many decomposition-based designs process raw components in parallel or use subbands directly as features, which can repeat low-frequency semantics across neighboring scales. JacoWaveNet instead builds separate approximation and detail residual pyramids: each finer level stores only the innovation over a wavelet-consistent lifted coarser level, and reconstruction follows classical coarse-to-fine wavelet synthesis (Daubechies, 1992; Mallat, 2008; Torrence & Compo, 1998). This residual view keeps stable trends reusable without repeatedly exposing the same signal at every scale.

2.3 Polynomial and Basis-Function Learning

Basis-function learning provides flexible nonlinear temporal responses. N-BEATS and N-HiTS use neural basis expansion or hierarchical interpolation, while KAN replaces fixed activations with learnable univariate functions and has inspired time-series and graph extensions (Oreshkin et al., 2020; Challu et al., 2023; Liu et al., 2024d; Huang et al., 2025; Li et al., 2025). TimeKAN and Wav-KAN further suggest that basis functions are most effective when the representation already exposes meaningful frequency or wavelet structure (Huang et al., 2025; Bozorgasl & Chen, 2024). Directly applying polynomial or spline mappings to raw sequences forces the basis to handle scale separation and nonlinear response modelling simultaneously. Our Jacobi Mapping module instead operates after dual-subband residual decomposition and after one signed temporal aggregation. Jacobi polynomials provide an orthogonal basis on the normalized message interval, and the maximum order increases the available element-wise response complexity; it is not a temporal-hop parameter or a graph-spectral filter. This separation lets the model assign scale decomposition, temporal routing, and nonlinear basis mapping to distinct stages.

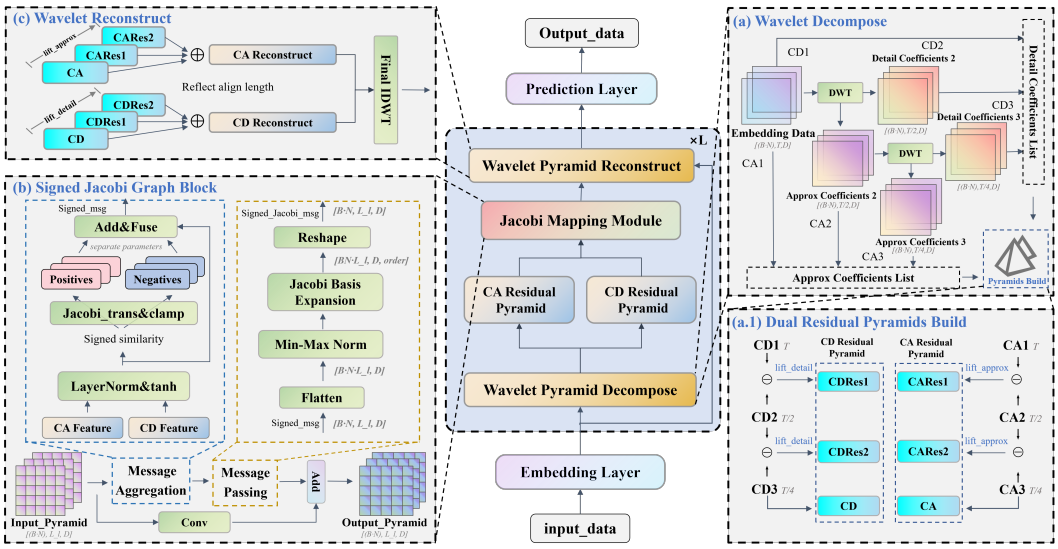


Figure 2: Overall architecture of JacoWaveNet. Panel (a) decomposes embedded sequences into approximation and detail coefficients by recursive DWT, and panel (a.1) builds dual residual pyramids to reduce cross-scale redundancy. Panel (b) details the Signed Jacobi Mapping block: one signed temporal aggregation is followed by an element-wise Jacobi basis mapping. Panel (c) reconstructs the two residual pyramids through wavelet-consistent coarse-to-fine lifting followed by one final IDWT.

3 METHOD

3.1 PROBLEM DEFINITION AND OVERALL ARCHITECTURE

Let $\mathbf{X} \in \mathbb{R}^{B\times T\times N}$ denote the observed multivariate sequence, where B, T, and N are the batch size, look-back length, and number of variables, respectively. The goal of time series forecasting is to predict

$$\hat{\mathbf{Y}} = f_\theta(\mathbf{X}) \in \mathbb{R}^{B\times P\times N}, \tag{1}$$

where P is the prediction length.

JacoWaveNet begins with a RevIN-style normalization on the original input (Kim et al., 2022). The normalized sequence is then rearranged into $B\cdot N$ single-variable streams,

$$\mathbf{X}_{\text{var}} \in \mathbb{R}^{BN\times T\times 1}, \tag{2}$$

A position-free embedding layer then maps each stream to the hidden representation:

$$\mathbf{Z}^0 = \text{Embed}(\mathbf{X}_{\text{var}}) \in \mathbb{R}^{BN\times T\times D}, \tag{3}$$

where D is the hidden dimension. The Jacobi Mapping (JM) module therefore learns temporal relations within each variable stream. This channel-independent design does not construct an explicit cross-variable relation graph; its scope is temporal modelling, while a cross-variable module is complementary.

The embedded sequence is processed by E JacoWaveNet blocks in succession. Let $\mathbf{Z}^{e-1}$ denote the representation entering block e; the block-level update is

$$\mathbf{Z}^e = \mathcal{B}^e(\mathbf{Z}^{e-1}), \qquad e = 1, \dots, E. \tag{4}$$

Each block follows the same data flow: wavelet decomposition, dual-subband wavelet residual pyramid construction, Jacobi Mapping (JM), and wavelet-consistent reconstruction. The dual-subband wavelet residual pyramid introduces structured multi-scale representations by separating approximation and detail components. JM then performs signed temporal aggregation followed by a nonlinear basis response in this decoupled representation space. After the final block, a temporal prediction head and a feature projection head produce the forecast:

$$\mathbf{O} = \text{Linear}_T\big((\mathbf{Z}^E)^\top\big)^\top \in \mathbb{R}^{BN\times P\times D}, \tag{5}$$

$$\hat{\mathbf{Y}} = \mathrm{Denorm}(\mathrm{Reshape}(\mathrm{Linear}_D(\mathbf{O}))) \in \mathbb{R}^{B\times P\times N}. \tag{6}$$

Here Linear_T maps the temporal dimension from T to P, and Linear_D maps the hidden dimension from D to 1 before reshaping back to the multivariate forecast.

3.2 Block-level pipeline

One JacoWaveNet block executes five steps. First, it recursively applies one-level DWT to obtain multi-level approximation/detail component pairs. Second, it residualizes approximation and detail components separately from coarse to fine, forming a dual-subband wavelet residual pyramid. Third, each residual level is refined by JM, which applies one signed temporal aggregation followed by an element-wise Jacobi basis mapping. Fourth, the approximation and detail residual pyramids are rebuilt from coarse to fine through wavelet-consistent lifting. Fifth, the block applies one final full IDWT on the finest reconstructed approximation/detail pair.

3.3 Wavelet analysis and synthesis operators

Each block contains one-step discrete wavelet analysis and synthesis operators (Daubechies, 1992; Mallat, 2008; Torrence & Compo, 1998),

$$(\mathbf{C}^A, \mathbf{C}^D) = \mathrm{DWT}(\mathbf{X}), \qquad \mathbf{X} = \mathrm{IDWT}(\mathbf{C}^A, \mathbf{C}^D), \tag{7}$$

where $\mathbf{C}^A$ and $\mathbf{C}^D$ denote approximation and detail components. The approximation component carries smoother low-frequency structure, while the detail component preserves localized variation at the current resolution. In the implementation, the default mother wavelet is `coif4`, and each transform step uses one decomposition level.

To preserve the synthesis semantics of the decimated wavelet hierarchy, cross-scale alignment is performed through valid wavelet lifting rather than direct coefficient interpolation:

$$\mathrm{Lift}_A(\mathbf{C}^A; \tau) = \mathrm{Align}\big(\mathrm{IDWT}(\mathbf{C}^A, \mathbf{0}), \tau\big), \tag{8}$$

$$\mathrm{Lift}_D(\mathbf{C}^D; \tau) = \mathrm{Align}\big(\mathrm{IDWT}(\mathbf{0}, \mathbf{C}^D), \tau\big), \tag{9}$$

where τ is the target temporal length. Small length mismatches caused by wavelet boundary handling are corrected by symmetric center cropping or reflect padding; linear interpolation is used only as a fallback for unexpectedly large gaps.

3.4 Dual-subband wavelet residual pyramid

Let $\mathbf{H} \in \mathbb{R}^{BN\times T\times D}$ be the input to a block. We recursively decompose its approximation coefficients over L wavelet scales. At each scale l, a one-level DWT produces an approximation/detail pair:

$$(\mathbf{C}_l^A, \mathbf{C}_l^D) = \mathrm{DWT}(\mathbf{C}_{l-1}^A), \qquad \mathbf{C}_0^A = \mathbf{H}, \quad l = 1, \ldots, L. \tag{10}$$

This produces a decimated wavelet tree in which only the approximation branch is further decomposed, while each level also yields a detail component.

For the approximation branch, the coarsest coefficient is kept as an anchor, and each finer coefficient is represented as the residual over the legally lifted coarser coefficient:

$$\mathbf{R}_L^A = \mathbf{C}_L^A, \tag{11}$$

$$\mathbf{R}_l^A = \mathbf{C}_l^A - \mathrm{Lift}_A\big(\mathbf{C}_{l+1}^A; |\mathbf{C}_l^A|\big), \qquad l = 1, \ldots, L-1. \tag{12}$$

For the detail branch,

$$\mathbf{R}_L^D = \mathbf{C}_L^D, \tag{13}$$

$$\mathbf{R}_l^D = \mathbf{C}_l^D - \mathrm{Lift}_D\big(\mathbf{C}_{l+1}^D; |\mathbf{C}_l^D|\big), \qquad l = 1, \ldots, L-1. \tag{14}$$

The resulting dual-subband wavelet residual pyramid is

$$\mathcal{P}_A = \{\mathbf{R}_1^A, \ldots, \mathbf{R}_{L-1}^A, \mathbf{R}_L^A\}, \qquad \mathcal{P}_D = \{\mathbf{R}_1^D, \ldots, \mathbf{R}_{L-1}^D, \mathbf{R}_L^D\}. \tag{15}$$

The coarsest nodes $\mathbf{R}_L^A$ and $\mathbf{R}_L^D$ are anchors rather than residuals in the usual sense. This design introduces structured multi-scale representations in two ways. First, approximation and detail components are modeled by two separate wavelet hierarchies rather than being merged too early. Second, each finer residual level mainly describes information not already explained by coarser content, reducing redundant low-frequency semantics and emphasizing scale-wise differential dynamics. These residual pyramids define the decoupled representation space used by the Jacobi Mapping (JM) module.

3.5 JACOBI MAPPING (JM) MODULE

The two residual pyramids are processed by independent branch processors,

$$\bar{\mathcal{P}}_A = \mathcal{F}_A(\mathcal{P}_A), \qquad \bar{\mathcal{P}}_D = \mathcal{F}_D(\mathcal{P}_D), \tag{16}$$

where $\mathcal{F}_A$ and $\mathcal{F}_D$ share the same architecture but do not share parameters. Each branch processor contains one JM module per pyramid level. Before and after the JM module, each level uses its own RevIN-style `Normalize` layer:

$$\bar{\mathbf{H}}_l^b = \mathrm{Denorm}_l^b\Big(\mathrm{JM}_l^b\Big(\mathrm{Norm}_l^b(\mathbf{H}_l^b)\Big)\Big), \qquad b \in \{A, D\}. \tag{17}$$

Given a normalized residual component $\mathbf{H}_l \in \mathbb{R}^{BN\times T_l\times D}$, JM first constructs a learned temporal-relation score:

$$\mathbf{S}_l = \tanh\big(\mathrm{LN}(\mathbf{H}_l)\,\mathrm{LN}(\mathbf{H}_l)^\top\big) \in \mathbb{R}^{BN\times T_l\times T_l}, \tag{18}$$

where LN denotes channel-wise layer normalization. Here, a *negative temporal relation* means $S_{l,ij} < 0$: a negative learned score between two time positions, rather than causal inhibition or necessarily a negative raw lag correlation. Let

$$\mathbf{U}_l = \alpha_{m(l)}\mathbf{S}_l - \beta_{m(l)}(\mathbf{I} - \mathbf{S}_l), \qquad \mathbf{A}_l = \mathrm{RowNorm}(\mathbf{U}_l), \tag{19}$$

where signed row normalization is defined as

$$\mathrm{RowNorm}(\mathbf{U})_{ij} = \frac{U_{ij}}{\sum_{r=1}^{T_l}|U_{ir}| + \epsilon}, \tag{20}$$

which preserves the sign of each entry. For $i \neq j$, $U_{l,ij} = (\alpha_{m(l)} + \beta_{m(l)})S_{l,ij}$, so the $\mathbf{I} - \mathbf{S}_l$ term does not reverse an off-diagonal negative relation; it only adjusts the diagonal before the explicit identity path below. The index $m(l)$ selects level-dependent coefficients,

$$m(l) = \begin{cases} c, & l = L, \\ h, & l < L, \end{cases} \tag{21}$$

where c denotes the coarsest level and h denotes higher-resolution levels. The final dependency map mixes this relation with the identity path:

$$\tilde{\mathbf{A}}_l = \eta_{m(l)}\mathbf{A}_l + \eta_o\mathbf{I}. \tag{22}$$

JM then separates positive and negative score magnitudes,

$$\tilde{\mathbf{A}}_l^+ = \max(\tilde{\mathbf{A}}_l, 0), \qquad \tilde{\mathbf{A}}_l^- = \max(-\tilde{\mathbf{A}}_l, 0). \tag{23}$$

The message used for polynomial filtering is:

$$\mathbf{M}_l = g_{\mathrm{self}}\mathbf{H}_l + g_{\mathrm{nb}}\left(g_+\tilde{\mathbf{A}}_l^+\mathbf{H}_l + g_-\tilde{\mathbf{A}}_l^-\mathbf{H}_l\right), \tag{24}$$

where g_{self}, g_{nb}, g_+, and g_- are learnable scalar gates. This is the only temporal aggregation in JM; the positive and negative branches preserve their magnitudes and are gated separately.

The aggregated message is min–max normalized channel-wise to $\widehat{\mathbf{M}}_{l,d} \in [-1, 1]$ and expanded on Jacobi polynomial bases, an orthogonal family on a bounded interval (Szegő, 1975). For channel $d \in \{1, \ldots, D\}$, let $\mathbf{M}_{l,d} = \mathbf{M}_l[:, :, d] \in \mathbb{R}^{BN\times T_l}$ denote the d-th channel. The expansion is applied element-wise after aggregation:

$$\Phi(\mathbf{M}_{l,d}) = \left[P_0^{(a,b)}(\widehat{\mathbf{M}}_{l,d}), P_1^{(a,b)}(\widehat{\mathbf{M}}_{l,d}), \ldots, P_K^{(a,b)}(\widehat{\mathbf{M}}_{l,d})\right]. \tag{25}$$

Here $\Phi(\mathbf{M}_{l,d})$ denotes the vector of Jacobi basis responses for channel d. The polynomial filtering response for channel d is:

$$\mathcal{J}(\mathbf{M}_{l,d}) = \sum_{k=0}^{K} w_{d,k}P_k^{(a,b)}(\widehat{\mathbf{M}}_{l,d}) + b_d. \tag{26}$$

The learned coefficients combine these element-wise basis responses. The maximum expansion order K controls nonlinear response complexity, not temporal-hop count, and $(a, b) = (0.5, 0.5)$ is the fixed symmetric setting. A lightweight depthwise temporal convolution is used as a local detail complement:

$$\mathbf{C}_l = \mathrm{DWConv1D}(\mathbf{H}_l). \tag{27}$$

The JM output is

$$\mathrm{JM}_l(\mathbf{H}_l) = \mathrm{Dropout}(\mathbf{W}_o\left(\mathbf{H}_l + \mathcal{J}(\mathbf{M}_l) + \mathbf{C}_l\right)). \tag{28}$$

Through this sequence, JM performs one signed temporal routing operation and a subsequent nonlinear Jacobi feature mapping. It is neither a polynomial of the relation matrix nor higher-order temporal message passing.

3.6 Coarse-to-fine reconstruction and forecasting

After JM refinement, the two residual pyramids are rebuilt independently from coarse to fine. For the approximation pyramid,

$$\hat{\mathbf{C}}_L^A = \bar{\mathbf{R}}_L^A, \tag{29}$$

$$\hat{\mathbf{C}}_l^A = \mathrm{Lift}_A\left(\hat{\mathbf{C}}_{l+1}^A;\, |\bar{\mathbf{R}}_l^A|\right) + \bar{\mathbf{R}}_l^A, \qquad l = L-1, \ldots, 1. \tag{30}$$

For the detail pyramid,

$$\hat{\mathbf{C}}_L^D = \bar{\mathbf{R}}_L^D, \tag{31}$$

$$\hat{\mathbf{C}}_l^D = \mathrm{Lift}_D\left(\hat{\mathbf{C}}_{l+1}^D;\, |\bar{\mathbf{R}}_l^D|\right) + \bar{\mathbf{R}}_l^D, \qquad l = L-1, \ldots, 1. \tag{32}$$

Only after both pyramids have been rebuilt does the block perform one full inverse wavelet transform on the finest reconstructed approximation/detail pair:

$$\hat{\mathbf{H}} = \mathrm{Align}\left(\mathrm{IDWT}\left(\hat{\mathbf{C}}_1^A, \hat{\mathbf{C}}_1^D\right), T\right). \tag{33}$$

Intermediate inverse transforms are used only as legal lifting operators for residual construction and reconstruction. The block output comes from one final IDWT, after which stacked blocks and the prediction head produce the final medium- or long-horizon forecast.

4 Experiments

Datasets and metrics. We evaluate on seven long-term forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity (ECL), and Traffic. We report horizons $\{96, 192, 336, 720\}$ and use MSE and MAE as the primary metrics. An external air-quality evaluation, data-generation details for the synthetic study, and detailed sensitivity, redundancy, scalability, and external-domain analyses are provided in Appendix Sec. F.

Baselines. We compare JacoWaveNet with representative baselines: TimeMosaic and SOFTS (multiscale/series mixing), iTransformer and CSFormer (transformer/channel models), and TQNet, DUET, ForecastGrapher, and MSGNet (relation-oriented models) (Ding et al., 2026; Lin et al., 2025; Wang et al., 2023; Qiu et al., 2025; Cai et al., 2024b; Han et al., 2024; Liu et al., 2024b; Cai et al., 2024a). Our runs use the stated common data pipeline when an implementation is available; otherwise, values are reproduced from the cited publication under its reported protocol. This provenance is recorded in Appendix Sec. B.

4.1 Main Results

Long-term Forecasting. Table 1 summarizes the average forecasting performance across four prediction horizons, while the complete horizon-wise results are reported in Appendix Table 8. Overall, JacoWaveNet is strongest or competitive across the evaluated benchmarks. It obtains the best average MSE on the ETT datasets and remains competitive on Weather, ECL, and Traffic. The controlled ablations, equal-order basis and fixed-wavelet sensitivities, and multi-seed stability check below test the individual design choices and robustness; the scale-redundancy diagnostic is reported in Appendix Sec. F. Missing baseline results are denoted by *N/A*.

Table 1: Average MSE and MAE over four forecasting horizons, corresponding to the Avg rows in Appendix Table 8. JacoWaveNet uses the reported seed-2021 configuration; baseline values follow the cited protocol as specified in Appendix Sec. B. The best and second-best entries are highlighted in red and blue, respectively.

Models	JacoWaveNet Ours		TimeMosaic 2026		TQNet 2025		CSFormer 2025		DUET 2025		ForecastGrapher 2024		SOFTS 2024		iTransformer 2024		MSGNet 2024	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	0.412	0.427	0.425	0.425	0.441	0.436	0.429	0.432	0.443	0.436	0.437	0.437	0.449	0.443	0.451	0.462	0.452	0.452
ETTh2	0.349	0.390	0.349	0.393	0.378	0.402	0.364	0.392	0.372	0.398	0.372	0.403	0.373	0.400	0.382	0.414	0.396	0.417
ETTm1	0.356	0.382	0.382	0.383	0.377	0.393	0.385	0.400	0.390	0.393	0.383	0.397	0.393	0.402	0.370	0.399	0.398	0.411
ETTm2	0.257	0.321	0.273	0.316	0.277	0.323	0.282	0.331	0.280	0.324	0.343	0.323	0.287	0.330	0.272	0.331	0.287	0.330
Weather	0.233	0.270	0.251	0.267	0.242	0.269	0.250	0.280	0.251	0.273	0.246	0.274	0.255	0.278	0.304	0.335	0.249	0.278
ECL	0.165	0.259	0.187	0.279	0.165	0.259	0.176	0.270	0.172	0.259	0.165	0.261	0.174	0.264	0.203	0.298	0.194	0.300
Traffic	0.413	0.286	0.458	0.283	0.445	0.276	*N/A*	*N/A*	0.451	0.269	0.458	0.292	0.409	0.267	0.397	0.295	0.641	0.370
1st Count	**6**	**3**	1	3	1	1	0	0	0	1	1	0	0	2	2	0	0	0

4.2 Ablation Study

Table 2 changes one component at a time on Weather and ETTm2 across four prediction horizons. Removing the residual pyramid raises average MSE by 4.2% and 7.5%, respectively; removing the negative-score branch raises it by 1.5% and 3.7%; and replacing JM raises it by 3.2% and 5.4%. These controlled results isolate the three components. The full-model rows are the same submitted records as the corresponding entries in the complete benchmark table (Appendix Table 8); the variants are separately trained controlled ablations. The direct approximation-branch redundancy diagnostic appears in Appendix Sec. F.

Table 2: Component ablation results on Weather and ETTm2. Results are reported as MSE/MAE.

Dataset	Model	96	192	336	720	Avg
Weather	JacoWaveNet	0.160/0.215	0.201/0.250	0.247/0.281	0.323/0.334	0.233/0.270
Weather	w/o Residual	0.171/0.222	0.215/0.259	0.261/0.293	0.323/0.339	0.242/0.278
Weather	w/o Negative	0.163/0.214	0.209/0.256	0.252/0.286	0.321/0.337	0.236/0.273
Weather	w/o Jacobi	0.166/0.217	0.212/0.258	0.258/0.290	0.325/0.340	0.240/0.276
ETTm2	JacoWaveNet	0.166/0.259	0.223/0.294	0.270/0.328	0.369/0.402	0.257/0.321
ETTm2	w/o Residual	0.178/0.268	0.242/0.304	0.296/0.339	0.389/0.397	0.276/0.327
ETTm2	w/o Negative	0.172/0.264	0.233/0.299	0.284/0.334	0.377/0.392	0.267/0.322
ETTm2	w/o Jacobi	0.175/0.266	0.237/0.301	0.289/0.337	0.382/0.395	0.271/0.325

4.3 Sensitivity Analysis

Equal-order basis and wavelet sensitivity. The polynomial map is applied to the bounded message domain $[-1, 1]$. Jacobi is orthogonal under $w_{\alpha,\beta}(x) = (1-x)^{\alpha}(1+x)^{\beta}$; the fixed $(\alpha, \beta) = (0.5, 0.5)$ setting is symmetric and does not introduce a second hand-designed sign offset after branch separation. All entries in Tables 3 and 4 use horizon 96. Their Jacobi/coif4 reference rows exactly match the corresponding 96-step JacoWaveNetentries in the complete benchmark table (Appendix Table 8). Table 3 compares equal-order bases. Their close performance is expected because they span closely related low-dimensional polynomial spaces; the comparison supports a range-matched parameterization, not a claim of greater polynomial capacity.

Table 3: Equal-order polynomial-basis sensitivity at horizon 96 (MSE/MAE; lower is better).

Basis	ETTm1	ETTm2	ETTh1	ETTh2	Weather
Jacobi	0.306/0.353	0.166/0.259	0.369/0.395	0.271/0.337	0.160/0.215
Legendre	0.307/0.356	0.171/0.261	0.368/0.396	0.276/0.339	0.162/0.209
Chebyshev-T	0.313/0.359	0.169/0.260	0.368/0.396	0.286/0.343	0.156/0.207
Chebyshev-U	0.315/0.358	0.173/0.263	0.369/0.398	0.281/0.340	0.160/0.212
Monomial	0.314/0.358	0.172/0.262	0.374/0.402	0.286/0.342	0.157/0.208

Table 4: Fixed-wavelet sensitivity at horizon 96 (MSE/MAE; lower is better). coif4 is the default; Haar and db4 are secondary sensitivity runs.

Wavelet	ETTm1	ETTm2	ETTh1	ETTh2	Weather
coif4	0.306/0.353	0.166/0.259	0.369/0.395	0.271/0.337	0.160/0.215
Haar	0.315/0.360	0.167/0.258	0.369/0.397	0.286/0.346	0.162/0.215
db4	0.315/0.354	0.168/0.259	0.369/0.397	0.288/0.346	0.160/0.216

coif4 has a near-symmetric 24-tap filter and 8/7 wavelet/scaling-function vanishing moments. It is a fixed default rather than a universally optimal wavelet; the sensitivity provides empirical support for this choice, especially on ETTh2, where coif4 obtains 0.271 MSE versus 0.286–0.288.

4.4 Stability Across Seeds

We retain the seed-2021 benchmark records as anchors and separately re-run three audited 96-step forecasts with five seeds. Table 5 shows low MSE variation: 1.257% (ETTm2), 2.796% (ETTh2), and 2.040% (Weather).

Table 5: Submitted seed-2021 anchors and separate five-seed re-runs at horizon 96. MSE/MAE are lower-is-better; CIs are 95% Student-t intervals.

Dataset	Anchor MSE / MAE	Re-run MSE (mean$\pm$SD; CI)	Re-run MAE (mean$\pm$SD; CI)
ETTm2	0.166 / 0.259	0.165$\pm$0.002; [0.162, 0.168]	0.256$\pm$0.003; [0.252, 0.260]
ETTh2	0.271 / 0.337	0.273$\pm$0.008; [0.263, 0.282]	0.339$\pm$0.004; [0.334, 0.344]
Weather	0.160 / 0.215	0.156$\pm$0.003; [0.152, 0.160]	0.212$\pm$0.005; [0.206, 0.218]

5 Limitations

JacoWaveNet targets regularly sampled multivariate series; irregular timestamps and long gaps require preprocessing or imputation. Dense within-subband relations scale quadratically with subband length, and the channel-independent design does not model cross-variable edges. The redundancy evidence concerns nested approximation coefficients only. Sparse routing, cross-variable modules, adaptive wavelets, and matched large-scale studies are natural extensions.

6 Conclusion

We propose JacoWaveNet, a dual-subband wavelet residual pyramid with signed polynomial temporal mapping for heterogeneous forecasting. It separates approximation and detail innovations, then applies one signed temporal aggregation and an element-wise Jacobi basis mapping. Multi-seed, sensitivity, redundancy, scalability, and external-domain analyses characterize its support and limits. Future work will explore sparse routing, cross-variable modelling, adaptive wavelets, and broader matched evaluations.

AI use statement

Generative AI assisted manuscript organization, editing, and reproducibility scripts only; it generated neither data nor experimental measurements. The authors reviewed all claims, configurations, and text, and take full responsibility for the manuscript.

Reproducibility statement

The paper and appendix specify the model, implementation, evaluation protocol, and hyperparameters. An anonymized code archive and experiment scripts will be supplied as supplementary material.

REFERENCES

Zavareh Bozorgasl and Hao Chen. Wav-KAN: Wavelet kolmogorov-arnold networks. *arXiv preprint arXiv:2405.12832*, 2024.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. MSGNet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024a.

Wanlin Cai, Kun Wang, Hao Wu, Xiaoxu Chen, and Yuankai Wu. ForecastGrapher: Redefining multivariate time series forecasting with graph neural networks. *arXiv preprint arXiv:2405.18036*, 2024b.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler-Canseco, and Artur Dubrawski. N-HiTS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023.

Song Chen. Beijing multi-site air-quality data. UCI Machine Learning Repository, 2019.

Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026. doi: 10.1609/aaai.v40i25.39218.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A unified multi-task time series model. In *Advances in Neural Information Processing Systems*, 2024.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the International Conference on Machine Learning*, 2024.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. SOFTS: Efficient multivariate time series forecasting with series-core fusion. In *Advances in Neural Information Processing Systems*, pp. 64145–64175, 2024. doi: 10.52202/079017-2046.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *Proceedings of the International Conference on Learning Representations*, 2024.

Ming Jin, Guangsi Shi, Yuan-Fang Li, Bo Xiong, Tian Zhou, Flora D. Salim, Liang Zhao, Lingfei Wu, Qingsong Wen, and Shirui Pan. Towards expressive spectral-temporal graph neural networks for time series forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *Proceedings of the International Conference on Learning Representations*, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

Maurice Kraus, Felix Divo, Devendra Singh Dhami, and Kristian Kersting. xLSTM-Mixer: Multivariate time series forecasting by mixing via scalar memories. *arXiv preprint arXiv:2410.16928*, 2024.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018.

Longlong Li, Yipeng Zhang, Guanghui Wang, and Kelin Xia. Kolmogorov–arnold graph neural networks for molecular property prediction. *Nature Machine Intelligence*, 7(8):1346–1354, 2025. doi: 10.1038/s42256-025-01087-7.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Proceedings of the International Conference on Machine Learning*, pp. 37797–37814, 2025.

Peiyuan Liu, Beiliang Wu, Naiqi Li, Tao Dai, Fengmao Lei, Jigang Bao, Yong Jiang, and Shu-Tao Xia. WFTNet: Exploiting global and local periodicity in long-term time series forecasting. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2024a.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2024b.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the International Conference on Machine Learning*, 2024c.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024d.

Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 2008.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the International Conference on Learning Representations*, 2023.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1185–1196, 2025. doi: 10.1145/3690624.3709325.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Yuxuan Shu and Vasileios Lampos. Sonnet: Spectral operator neural network for multivariable time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25419–25427, 2026. doi: 10.1609/aaai.v40i30.39736.

Gábor Szegő. *Orthogonal Polynomials*. American Mathematical Society, fourth edition, 1975.

Christopher Torrence and Gilbert P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 1998.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Haoxin Wang, Yipeng Mo, Kunlan Xiang, Nan Yin, Honghe Dai, Bixiong Li, Songhai Fan, and Site Mo. CSformer: Combining channel independence and mixing for robust multivariate time series forecasting. *arXiv preprint arXiv:2312.06220*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. TimeMixer++: A general time series pattern machine for universal predictive analysis. In *Proceedings of the International Conference on Learning Representations*, 2025.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the International Conference on Machine Learning*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations*, 2023.

Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. *arXiv preprint arXiv:2510.14510*, 2025.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 753–763, 2020. doi: 10.1145/3394486.3403118.

Jiyuan Xu, Wenyu Zhang, Xin Jing, Jiahao Nie, Shuai Chen, and Shuai Zhang. CPiRi: Channel permutation-invariant relational interaction for multivariate time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2026.

Fuhao Yang, Xin Li, Min Wang, Hongyu Zang, Wei Pang, and Mingzhong Wang. WaveForM: Graph enhanced wavelet learning for long sequence forecasting of multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Xian Yang, Zhenguo Zhang, and Shihao Lu. MSCGrapher: Learning multi-scale dynamic correlations for multivariate time series forecasting. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 4445–4456. PMLR, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the International Conference on Machine Learning*, 2022.

A Societal Impact

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically singled out beyond the general considerations below.

Our research advances machine learning by enhancing the capabilities of long-term forecasting in time series models, improving both accuracy and efficiency for heterogeneous temporal dynamics. By developing JacoWaveNet, we introduce a forecasting framework that can be applied across domains such as finance, healthcare, energy, transportation, and logistics. Improved forecasting accuracy may support better decision-making in areas such as resource allocation, demand planning, risk management, and early warning systems.

However, we also recognize potential risks associated with the misuse of advanced forecasting models. Time series forecasting models could be leveraged for harmful purposes, especially when applied at scale. For example, in financial settings, adversarial agents might use forecasts to support manipulative trading strategies or amplify market instability. In political or social contexts, forecasting tools could be combined with other systems to anticipate and influence public behavior. In sensitive domains such as healthcare, energy, and security, overreliance on inaccurate forecasts or poorly calibrated deployments may lead to unintended consequences.

Therefore, it is essential that the use of JacoWaveNet, like all machine learning technologies, is guided by responsible practices and ethical considerations. We encourage stakeholders to adopt rigorous evaluation processes to ensure fairness, transparency, robustness, and accountability in deployment, and to remain vigilant to the broader societal implications of time series forecasting technologies.

B Implementation Details

Experimental Details. Our codebase is implemented in Python using PyTorch for model training and optimization (Paszke et al., 2019). JacoWaveNet uses `pytorch_wavelets` for one-dimensional DWT and IDWT operators, and relies on NumPy, pandas, scikit-learn, and Matplotlib for preprocessing, metric aggregation, and visualization. All experiments were run on a server equipped with a single NVIDIA RTX A6000 GPU with CUDA 11.8. Data loading and preprocessing follow the standard long-term forecasting protocol: ETT datasets use the established train/validation/test chronological splits, while custom datasets such as Weather, ECL, and Traffic use a 70%/10%/20% chronological split. All reported long-term forecasting experiments use multivariate-to-multivariate forecasting with horizons $\{96, 192, 336, 720\}$ and input length 512.

The released repository contains the training entry point `run.py`, benchmark scripts such as `scripts/long.sh`, ablation scripts, and the Jacobi-order sensitivity script `scripts/run_jacobi_order_sensitivity.py`. Baseline implementations available in the repository are run through the same data-loading and evaluation pipeline when applicable; remaining comparison values are taken from the corresponding published reports.

Training and Hyperparameters. We optimize JacoWaveNet in 32-bit precision with the Adam optimizer (Kingma & Ba, 2015). The default training objective is MSE loss, and performance is reported with MSE and MAE. We use a OneCycle learning-rate schedule, early stopping on validation loss, and fixed random seeds for Python, NumPy, and PyTorch. Unless otherwise stated, the main experiments use seed 2021, two JacoWaveNet blocks, model dimension 16, dropout 0.1, two wavelet down-sampling layers, and a down-sampling window of 2.

Table 6: Main implementation and training configuration used by JacoWaveNet.

Hyperparameter	Value
Compute hardware	NVIDIA RTX A6000
CUDA version	11.8
Input length	512
Forecast horizons	96/192/336/720
Model dimension	16
Number of JacoWaveNet blocks	2
Dropout	0.1
Wavelet down-sampling layers	2
Wavelet down-sampling window	2
Optimizer	Adam
Learning-rate schedule	OneCycleLR
Training epochs	10
Early-stopping patience	10
Main benchmark batch size	4
Random seed	2021

C Wavelet Decomposition and Jacobi Basis Visualization

Wavelet decomposition. The wavelet residual pyramid in JacoWaveNet starts from a decimated discrete wavelet transform. Let $\mathbf{C}_0^A = \mathbf{H}$ denote the embedded sequence entering one block. With low-pass filter h and high-pass filter g, one analysis step can be written as

$$\mathbf{C}_l^A[t,d] = \sum_s h[s]\mathbf{C}_{l-1}^A[2t-s,d], \qquad \mathbf{C}_l^D[t,d] = \sum_s g[s]\mathbf{C}_{l-1}^A[2t-s,d], \tag{34}$$

where $l = 1, \ldots, L$ and d indexes the hidden channel. The approximation component $\mathbf{C}_l^A$ keeps the smoother low-frequency content at level l, while the detail component $\mathbf{C}_l^D$ captures local high-frequency deviations removed by the low-pass branch. As in a standard multiresolution analysis, only the approximation branch is recursively decomposed, producing one approximation and one detail component at each level.

Instead of forwarding these raw coefficients directly, JacoWaveNet forms residual pyramids for both subbands. Each finer coefficient is compared with the wavelet-consistent lift of its coarser counterpart:

$$\mathbf{R}_l^A = \mathbf{C}_l^A - \mathrm{Lift}_A\big(\mathbf{C}_{l+1}^A; |\mathbf{C}_l^A|\big), \qquad \mathbf{R}_l^D = \mathbf{C}_l^D - \mathrm{Lift}_D\big(\mathbf{C}_{l+1}^D; |\mathbf{C}_l^D|\big). \tag{35}$$

This residualization removes repeated low-frequency semantics across neighboring scales and makes each level emphasize the information newly introduced at that resolution. Reconstruction follows the reverse coarse-to-fine process and applies the final inverse DWT only after both approximation and detail pyramids have been rebuilt.

Jacobi basis comparison. The Jacobi Mapping module expands normalized messages on Jacobi polynomial bases (Szegő, 1975). For parameters α and β, the first two bases are

$$P_0^{(\alpha,\beta)}(x) = 1, \qquad P_1^{(\alpha,\beta)}(x) = \frac{1}{2}\left((2+\alpha+\beta)x + \alpha - \beta\right). \tag{36}$$

For $k \geq 2$, the higher-order terms are generated recursively:

$$\begin{aligned} P_k^{(\alpha,\beta)}(x) &= \frac{a_{2,k}a_{3,k}(x)P_{k-1}^{(\alpha,\beta)}(x) - a_{4,k}P_{k-2}^{(\alpha,\beta)}(x)}{a_{1,k}}, \\ a_{1,k} &= 2k(k+\alpha+\beta)(2k+\alpha+\beta-2), \\ a_{2,k} &= 2k+\alpha+\beta-1, \\ a_{3,k}(x) &= (2k+\alpha+\beta)(2k+\alpha+\beta-2)x + \alpha^2 - \beta^2, \\ a_{4,k} &= 2(k+\alpha-1)(k+\beta-1)(2k+\alpha+\beta). \end{aligned} \tag{37}$$

Figure 3 visualizes the resulting basis family for orders $k = 0, \ldots, 6$ under the neutral setting $\alpha = \beta = 0.5$ used in the implementation. Lower orders vary smoothly over the normalized interval

$[-1, 1]$, while higher orders introduce more zero crossings and sharper local responses. This explains why increasing K mainly changes the model's sensitivity to localized residual patterns rather than simply increasing capacity monotonically.

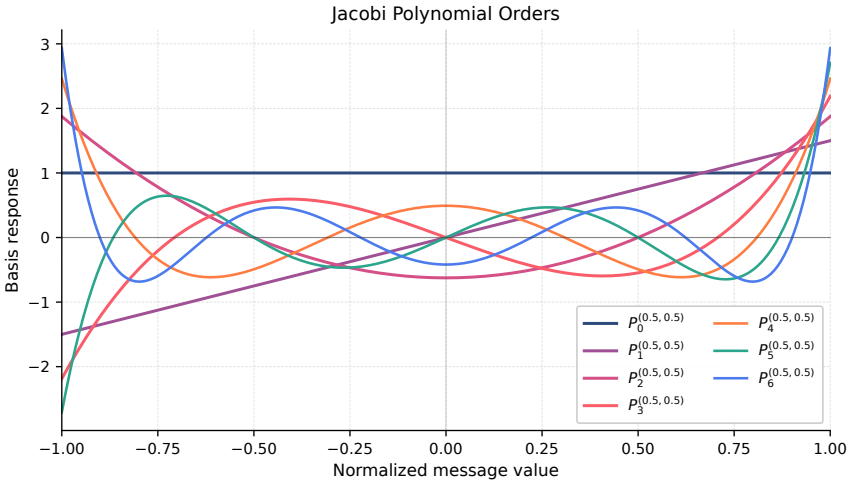


Figure 3: Jacobi polynomial basis responses for orders $k = 0, \ldots, 6$ with neutral parameters $\alpha = \beta = 0.5$. The low-order curves provide smooth global responses, while higher-order curves introduce increasingly oscillatory local responses for residual subband modeling.

D Signed Temporal Relation Semantics

This appendix records the intended scope of the signed relation module. For each variable stream, JM learns relations between time positions only; it does not construct cross-variable edges. A negative relation is a negative learned score between two time positions, rather than a causal-inhibition claim or a raw negative lag correlation.

The signed aggregation in the main paper preserves the sign of every off-diagonal learned score before normalization. The subsequent Jacobi expansion is applied element-wise to normalized aggregate messages. Consequently, the Jacobi order controls response complexity after temporal routing; it is neither a relation-matrix polynomial nor a temporal-hop count.

E Reproducibility and Scope

The main manuscript and Appendix B specify the model, optimization configuration, random seed, data splits, and evaluation metrics. The experimental materials in Appendix F distinguish the submitted seed-2021 benchmark records from separate five-seed re-runs, so the two should not be conflated.

The method targets regularly sampled multivariate sequences and uses dense within-subband temporal relations. It does not claim explicit cross-variable modelling, universal wavelet optimality, or a redundancy result for arbitrary detail-band pairs. These scope boundaries are stated in the main-paper limitations and supported by the targeted diagnostics in Appendix F.

F Supplementary Experiments

F.1 Benchmark Setting and Complete Results

Dataset details. Table 7 provides an overview of the benchmark datasets used to evaluate JacoWaveNet under the standard long-term forecasting protocol. The four ETT datasets contain electricity transformer temperature measurements from two transformers at different temporal resolutions:

ETTh1 and ETTh2 are hourly series, while ETTm1 and ETTm2 are recorded every 15 minutes. Weather contains 21 meteorological variables collected at a high temporal frequency, making it useful for evaluating short-term fluctuations together with longer seasonal patterns. ECL records electricity consumption from 321 clients, and Traffic contains road occupancy measurements from 862 sensors, both of which test scalability to high-dimensional multivariate forecasting. Following common practice in long-term time series forecasting (Zhou et al., 2021; Wu et al., 2021), all datasets are evaluated with prediction horizons $\{96, 192, 336, 720\}$ using MSE and MAE.

Table 7: Benchmark datasets used in our experiments.

Dataset	Domain	#Variables	Horizons
ETTh1	Electricity transformer temperature	7	96/192/336/720
ETTh2	Electricity transformer temperature	7	96/192/336/720
ETTm1	Electricity transformer temperature	7	96/192/336/720
ETTm2	Electricity transformer temperature	7	96/192/336/720
Weather	Meteorology	21	96/192/336/720
ECL	Power consumption	321	96/192/336/720
Traffic	Road occupancy	862	96/192/336/720

Evaluation metrics. Following common practice in long-term forecasting (Zhou et al., 2021; Wu et al., 2021), we evaluate deterministic forecasts with Mean Squared Error (MSE) and Mean Absolute Error (MAE). Metrics are computed for each variate and then averaged over all variates. Both metrics are lower-is-better, and all reported averages over horizons are arithmetic means of the corresponding horizon-wise scores.

Full long-term forecasting results.

Table 8: Mean Squared Error (MSE) and Mean Absolute Error (MAE) on seven real-world datasets. The best results are highlighted in red, and the second-best results are highlighted in blue. Avg represents the average results across the four forecasting horizons. The final row reports the number of first-place entries, with ties counted separately.

	Metrics	JacoWaveNet		TimeMosaic 2026		TQNet 2025		CSFormer 2025		DUET 2025		ForecastGrapher 2024		SOFTS 2024		iTransformer 2024		MSGNet 2024	
Datasets	**Horizon**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ETTh1	96	0.369	0.395	0.369	0.389	0.371	0.393	0.372	0.394	0.377	0.393	0.373	0.397	0.381	0.399	0.395	0.420	0.390	0.411
	192	0.401	0.415	0.416	0.415	0.428	0.426	0.420	0.425	0.429	0.425	0.424	0.427	0.435	0.431	0.427	0.441	0.442	0.442
	336	0.428	0.437	0.454	0.439	0.476	0.456	0.453	0.440	0.471	0.446	0.472	0.448	0.480	0.452	0.445	0.457	0.480	0.468
	720	0.449	0.461	0.461	0.458	0.487	0.470	0.470	0.470	0.496	0.480	0.479	0.475	0.499	0.488	0.537	0.530	0.494	0.488
	Avg	0.412	0.427	0.425	0.425	0.441	0.436	0.429	0.432	0.443	0.436	0.437	0.437	0.449	0.443	0.451	0.462	0.452	0.452
ETTh2	96	0.271	0.337	0.286	0.339	0.295	0.343	0.293	0.340	0.296	0.345	0.294	0.345	0.297	0.347	0.304	0.360	0.328	0.371
	192	0.335	0.378	0.362	0.380	0.367	0.393	0.375	0.390	0.368	0.389	0.367	0.396	0.373	0.394	0.377	0.403	0.402	0.414
	336	0.362	0.406	0.409	0.416	0.417	0.427	0.378	0.406	0.411	0.422	0.407	0.428	0.410	0.426	0.405	0.429	0.435	0.443
	720	0.399	0.440	0.396	0.437	0.433	0.446	0.409	0.432	0.412	0.434	0.420	0.441	0.411	0.433	0.443	0.464	0.417	0.441
	Avg	0.349	0.390	0.349	0.393	0.378	0.402	0.364	0.392	0.372	0.398	0.372	0.403	0.373	0.400	0.382	0.414	0.396	0.417
ETTm1	96	0.306	0.353	0.308	0.338	0.311	0.353	0.324	0.367	0.324	0.354	0.317	0.357	0.325	0.361	0.312	0.366	0.319	0.366
	192	0.337	0.373	0.364	0.375	0.356	0.378	0.369	0.388	0.369	0.379	0.365	0.383	0.375	0.389	0.347	0.385	0.376	0.397
	336	0.366	0.385	0.394	0.390	0.390	0.401	0.396	0.408	0.404	0.402	0.393	0.405	0.405	0.412	0.379	0.404	0.417	0.422
	720	0.413	0.418	0.460	0.428	0.452	0.440	0.451	0.439	0.463	0.437	0.456	0.442	0.466	0.447	0.441	0.442	0.481	0.458
	Avg	0.356	0.382	0.382	0.383	0.377	0.393	0.385	0.400	0.390	0.393	0.383	0.397	0.393	0.402	0.370	0.399	0.398	0.411
ETTm2	96	0.166	0.259	0.170	0.248	0.173	0.256	0.170	0.269	0.174	0.255	0.442	0.259	0.180	0.261	0.179	0.271	0.177	0.262
	192	0.223	0.294	0.236	0.299	0.238	0.298	0.244	0.309	0.243	0.302	0.238	0.300	0.246	0.306	0.242	0.313	0.247	0.307
	336	0.270	0.328	0.295	0.331	0.301	0.340	0.303	0.346	0.304	0.341	0.296	0.338	0.319	0.352	0.288	0.344	0.312	0.346
	720	0.369	0.402	0.392	0.386	0.397	0.396	0.400	0.400	0.399	0.397	0.395	0.396	0.405	0.401	0.378	0.397	0.414	0.403
	Avg	0.257	0.321	0.273	0.316	0.277	0.323	0.282	0.331	0.280	0.324	0.343	0.323	0.287	0.330	0.272	0.331	0.287	0.330
Weather	96	0.160	0.215	0.165	0.198	0.157	0.200	0.168	0.217	0.163	0.202	0.161	0.206	0.166	0.208	0.253	0.304	0.163	0.212
	192	0.201	0.250	0.215	0.243	0.206	0.245	0.213	0.257	0.218	0.252	0.209	0.251	0.217	0.253	0.280	0.319	0.212	0.254
	336	0.247	0.281	0.272	0.286	0.262	0.287	0.272	0.298	0.274	0.294	0.268	0.295	0.282	0.300	0.321	0.344	0.272	0.299
	720	0.323	0.334	0.353	0.340	0.344	0.342	0.346	0.347	0.349	0.343	0.348	0.345	0.356	0.351	0.364	0.374	0.350	0.348
	Avg	0.233	0.270	0.251	0.267	0.242	0.269	0.250	0.280	0.251	0.273	0.246	0.274	0.255	0.278	0.304	0.335	0.249	0.278
ECL	96	0.137	0.234	0.160	0.258	0.134	0.229	0.146	0.242	0.145	0.233	0.138	0.235	0.143	0.233	0.147	0.248	0.165	0.274
	192	0.151	0.246	0.174	0.269	0.154	0.247	0.172	0.266	0.163	0.248	0.154	0.249	0.158	0.248	0.165	0.267	0.184	0.292
	336	0.167	0.262	0.189	0.283	0.169	0.264	0.176	0.271	0.175	0.262	0.169	0.264	0.178	0.269	0.178	0.279	0.195	0.302
	720	0.205	0.295	0.225	0.304	0.201	0.294	0.211	0.303	0.204	0.291	0.199	0.294	0.218	0.305	0.322	0.398	0.231	0.332
	Avg	0.165	0.259	0.187	0.279	0.165	0.259	0.176	0.270	0.172	0.259	0.165	0.261	0.174	0.264	0.203	0.298	0.194	0.300
Traffic	96	0.392	0.276	0.424	0.269	0.413	0.261	*N/A*	*N/A*	0.407	0.252	0.423	0.278	0.376	0.251	0.367	0.288	0.605	0.344
	192	0.402	0.279	0.448	0.279	0.432	0.271	*N/A*	*N/A*	0.431	0.262	0.445	0.285	0.398	0.261	0.378	0.293	0.613	0.359
	336	0.411	0.283	0.464	0.284	0.450	0.277	*N/A*	*N/A*	0.456	0.269	0.460	0.292	0.415	0.269	0.399	0.294	0.642	0.376
	720	0.448	0.304	0.495	0.301	0.486	0.295	*N/A*	*N/A*	0.509	0.292	0.503	0.312	0.447	0.287	0.442	0.304	0.702	0.401
	Avg	0.413	0.286	0.458	0.283	0.445	0.276	*N/A*	*N/A*	0.451	0.269	0.458	0.292	0.409	0.267	0.397	0.295	0.641	0.370
1st Count		**26**	**17**	3	11	3	2	0	2	0	5	2	0	0	10	10	0	0	0

F.2 Qualitative and Synthetic Evaluation

Prediction visualization.

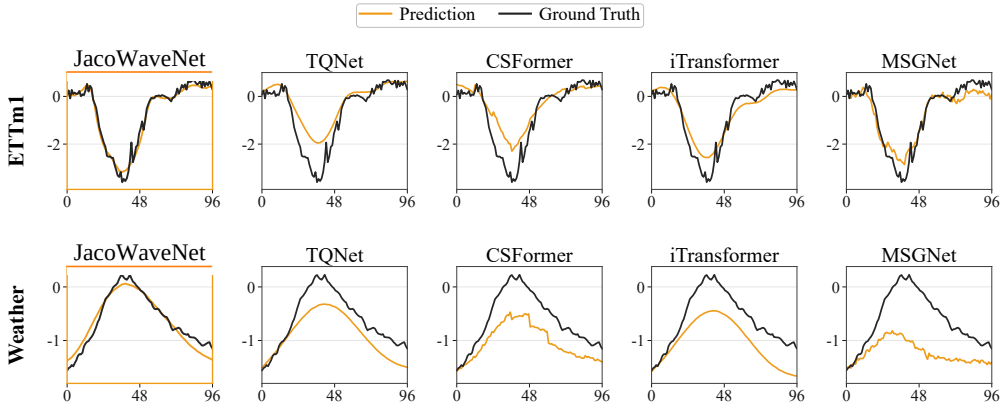


Figure 4: Visualization of prediction results on the ETTm1 and Weather datasets with input length 512 and output length 96. The horizontal axis denotes the forecasting horizon, and the vertical axis denotes the normalized value of the selected variable; the orange and black curves represent prediction and ground truth, respectively.

Synthetic heterogeneity generator. The coefficient $h \in [0, 1]$ is a dimensionless stress variable, not a physical measurement. For a fixed seed, the generator is

$$b_h(t) = s(t) + (0.10 + 0.55h)m(t) + (0.02 + 0.40h)f(t) + \ell_h(t), \tag{38}$$

$$x_h(t) = b_h(t) - (0.05 + 0.65h)b_h(t - \tau) + 0.25h\, r_h(t)f(t) + \epsilon_h(t), \tag{39}$$

where s, m, f are slow, medium, and fast components; ℓ_h is an increasingly frequent and large local-event process; r_h switches signs; and $\epsilon_h(t) \sim \mathcal{N}(0, (0.04+0.16h)^2)$. Increasing h jointly strengthens medium/fast variation, local events, sign-switching interactions, and noise. **Synthetic heterogeneity analysis.** Beyond the real-world benchmark results above, we further examine robustness under controlled heterogeneous temporal patterns. Table 9 reports results on synthetic data with varying heterogeneity coefficients. As the coefficient increases from $h = 0.0$ to $h = 1.0$, all methods become less accurate, reflecting the increasing difficulty of the forecasting task. However, the performance gap between the full model and its ablated variants consistently widens under stronger heterogeneity. This trend suggests that JacoWaveNet is better able to handle heterogeneous temporal structures, which is consistent with its explicit decomposition of subband dynamics and dependency modeling in a decoupled representation space.

Table 9: Synthetic heterogeneity analysis under different heterogeneity coefficients. Results are reported as MSE/MAE.

Horizon	Model	$h = 0.0$	$h = 0.5$	$h = 1.0$	Avg
96	JacoWaveNet	0.082/0.168	0.101/0.189	0.128/0.214	0.104/0.190
96	w/o Residual	0.090/0.176	0.117/0.201	0.152/0.231	0.120/0.203
96	w/o Negative	0.087/0.173	0.111/0.197	0.143/0.224	0.114/0.198
96	w/o Jacobi	0.089/0.175	0.114/0.199	0.148/0.228	0.117/0.201
336	JacoWaveNet	0.121/0.218	0.156/0.247	0.203/0.286	0.160/0.250
336	w/o Residual	0.133/0.229	0.171/0.261	0.225/0.304	0.176/0.265
336	w/o Negative	0.128/0.225	0.165/0.256	0.216/0.297	0.170/0.259
336	w/o Jacobi	0.130/0.227	0.168/0.259	0.220/0.301	0.173/0.262

F.3 Controlled Ablations and Sensitivity Analyses

Jacobi order sensitivity. Complementing the basis-level comparison in Figure 3, Table 10 summarizes the sensitivity of JacoWaveNet to the Jacobi polynomial order K on ETTm2. This analysis should be interpreted together with the wavelet residual representation: low-order bases provide smooth responses that are naturally aligned with approximation residuals and low-frequency trends, whereas higher-order bases add oscillatory responses that can better match detail residuals and local high-frequency deviations. Since K is the maximum expansion order, increasing it augments the available basis set rather than replacing low-order responses; the learned filtering weights can still emphasize the lower-order terms when the horizon or subband is dominated by smooth structure.

The results are therefore horizon-dependent rather than monotonic. Lower orders remain competitive on horizons where smooth approximation information dominates, while higher orders can help when localized detail responses are useful. However, very high orders may also introduce unnecessary sensitivity to noisy residual fluctuations. The average MSE/MAE varies within a narrow range across $K = 1$ to $K = 6$, indicating that the proposed residual decomposition makes the model robust to the exact choice of the maximum Jacobi order.

Table 10: Sensitivity analysis of the Jacobi polynomial order K on ETTm2. Results are reported as MSE/MAE; lower is better. Avg averages over the four forecasting horizons.

K	96	192	336	720	Avg
1	0.174/0.265	0.221/0.297	0.273/0.329	0.363/0.393	0.257/0.321
2	0.171/0.261	0.247/0.321	0.278/0.337	0.360/0.392	0.264/0.328
3	0.169/0.263	0.234/0.314	0.282/0.337	0.362/0.395	0.262/0.327
4	0.172/0.265	0.221/0.297	0.275/0.337	0.366/0.394	0.259/0.323
5	0.175/0.267	0.226/0.308	0.275/0.332	0.360/0.391	0.259/0.324
6	0.167/0.258	0.227/0.302	0.275/0.341	0.364/0.390	0.258/0.323

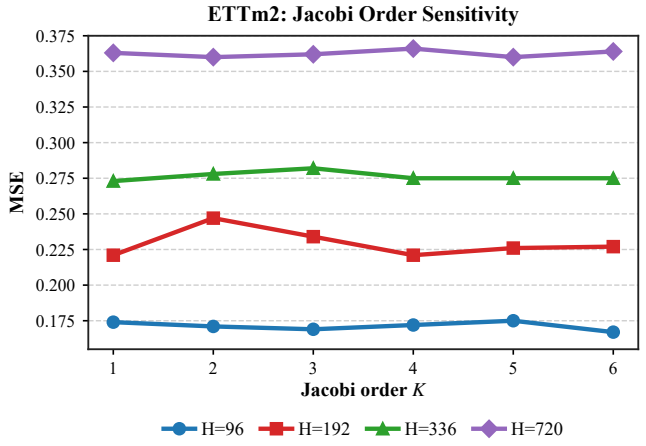


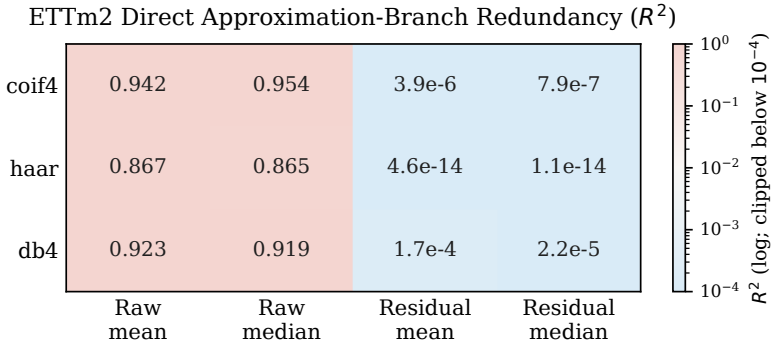
Figure 5: MSE sensitivity of JacoWaveNet to the Jacobi polynomial order K on ETTm2. Varying K changes the balance between smooth low-order responses and localized high-order responses, so the effect is horizon-dependent rather than monotonic.

F.4 Redundancy, Scalability, and External Validation

Direct approximation-branch redundancy diagnostic. For 1,024 ETTm2 test windows and seven variables (7,168 coefficient streams), we measure squared Pearson correlation between a fine approximation and the wavelet-consistent lift of its adjacent coarse approximation, before and after residualization. Table 11 reports the result. The direct redundancy claim concerns nested approximation coefficients. Adjacent detail bands are non-nested band-pass components; after subtraction, their correlations can change algebraically and are not used as a redundancy metric.

Table 11: Approximation-branch adjacent-scale diagnostic on ETTm2.

Wavelet	Raw R^2	Residual R^2	Residual / fine energy
coif4	0.942	< 0.001	1.611%
Haar	0.867	< 0.001	1.827%
db4	0.923	< 0.001	1.582%



Each cell is a direct mean or median over 7,168 coefficient streams; annotations retain unclipped values.

Figure 6: Heatmap from direct ETTm2 approximation-branch diagnostics (1,024 test windows; 7,168 coefficient streams per wavelet). Cells report the measured mean or median squared Pearson correlation before and after residualization. Colors use a log scale clipped below 10^{-4} for visibility, while annotations retain the direct, unclipped values.

Scalability and external-domain validation. With subband lengths T_l, embedding dimension D, blocks E, batch size B, and N variables, dense temporal relation computation costs $O(BNED\sum_l T_l^2)$. Parameter count is weakly dependent on N, but latency and activation memory grow with the number of variable streams. Table 12 makes this trade-off explicit.

Table 12: Batch-1 inference profile (input 512, horizon 96; NVIDIA P40).

Variables N	Parameters	Latency	Peak memory
7	53,031	11.300 ms	17.820 MiB
21	53,059	12.220 ms	35.520 MiB
321	53,659	121.170 ms	418.180 MiB
862	54,741	321.070 ms	1,100.720 MiB

We additionally evaluate AQWan, the 11-variable Wanshouxigong subset of the Beijing Multi-Site Air-Quality data (Wu et al., 2025; Chen, 2019). With input length 512, chronological 7:1:2 splitting, seed 2021, Jacobi mapping, and coif4, the model obtains the following external-domain results.

Table 13: AQWan external-domain results (MSE/MAE/RMSE; lower is better).

Horizon	MSE	MAE	RMSE
96	0.708	0.463	0.842
192	0.759	0.484	0.871
336	0.784	0.494	0.885
720	0.856	0.524	0.925