SPECTRUM-DUALNET: SPECTRUM-GUIDED ROUTING OF TIME-FREQUENCY DUAL-NETWORK VIA ADAPTIVE DECOMPOSITION FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series contain temporal patterns evolving at timescales with different characteristics like days or week periods, resulting in entangled long-term trends and short-term oscillations. Spectral information helps disentangle these heterogeneous temporal dynamics by distinguishing low-frequency trends from high-frequency fluctuations, thereby facilitating more effective forecasting. However, widely used trend-seasonal decomposition methods typically rely on predefined decomposition structures and are not designed to adaptively identify multiple spectrum modes with distinct central frequencies across different datasets. Yet, how to properly disentangle informative temporal patterns and route them to spectrum-specialized branches according to their spectra remains underexplored in time series forecasting. To address this, we propose Spectrum-DualNet, a dual-branch forecasting network that adaptively separates multivariate signals according to the distributions of their datasets into band-limited modes that belong to distinct frequency ranges as characteristic spectra using Multivariate Variational Mode Decomposition (MVMD). The lowest-frequency mode is routed to a trend branch in the time domain to capture long-term dependencies, while the remaining high-frequency modes are routed to a fluctuation branch that models non-stationary dynamics in the frequency domain. The two branches are fused through a gating network which computes mixing coefficients that weight the trend and fluctuation components. The decomposition residual is incorporated through a second gate, allowing information not captured by the extracted modes to contribute to forecasting. Experiments on real-world time series datasets demonstrate that Spectrum-DualNet achieves competitive forecasting performance.

1 INTRODUCTION

Time series forecasting provides essential information for decision-making across a wide range of applications (Kim et al., 2025), where future observations depend on interacting temporal patterns at different timescales (Wu et al., 2021). Real-world time series often contain slowly varying trends and oscillatory fluctuations whose characteristics evolve over time (Huang et al., 2025). With high-frequency components capturing short-term fluctuations and low-frequency components reflecting long-term trends as the simple sample shown in Figure 1. These patterns can be distinguished by their spectral characteristics in the frequency domain, providing a basis for differentiating temporal dynamics. Such spectral differences also motivate distinct modeling strategies: slowly varying components benefit from representations of long-term temporal structure, whereas oscillatory components call for sensitivity to short-term amplitude and phase fluctuations. Spectral information therefore offers a common basis for both separating temporal components and guiding their assignment to specialized forecasting branches.

Decomposition and frequency filtering provide natural starting points for disentangling complex temporal patterns. FreqCycle combines learnable periodic templates based on a predefined period (Zhang et al., 2026). TimeKAN constructs a hierarchy through moving-average downsampling and

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.

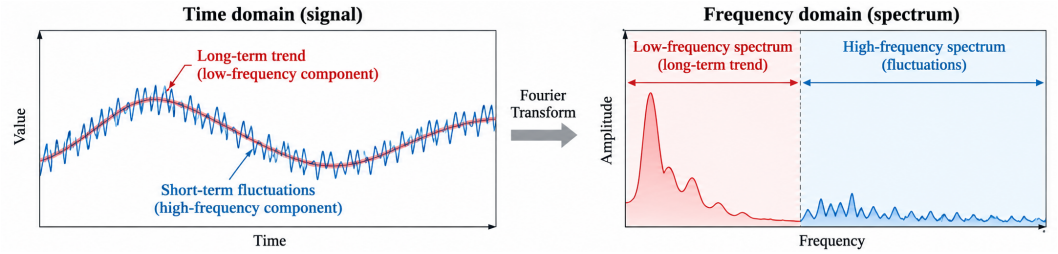


Figure 1: Illustration of the correspondence between time-domain dynamics and spectral characteristics, where low-frequency components represent the long-term trend and high-frequency components capture short-term fluctuations.

extracts representations through cascaded decomposition (Huang et al., 2025), while TimeMixer mixes seasonal and trend representations across predefined downsampling scales (Wang et al., 2024).

These advances demonstrate the value of multiscale decomposition and adaptive spectral processing. However, their temporal organization relies partly on predefined periods or scales, so their adaptability to different datasets is limited. Moreover, although moving-average filtering is simple and widely used in time series decomposition, its limited frequency selectivity makes it an inefficient filter in the frequency domain (Smith, 1997). Having observed these issues of frequency decomposition, we propose a complementary decomposition strategy that adaptively estimates characteristic frequencies of multivariate data according to the distributions of their datasets, and uses the resulting modes as a consistent decomposition basis.

Beyond separating temporal components, forecasting effectiveness also depends on how the extracted components are modeled. Their spectral characteristics provide a basis for this choice: lower-frequency modes describe slower temporal evolution, whereas higher-frequency modes contain faster oscillations. Recent methods model different time series components within a unified modeling framework: TimeKAN assigns different KAN orders to the extracted components (Huang et al., 2025), Amplifier uses separate feed-forward predictors for seasonal and trend components (Fei et al., 2025), and FilterNet applies learnable frequency filters before forecasting from the representation (Yi et al., 2024).

Although these approaches account for spectral heterogeneity, using a common modeling mechanism across components limits their ability to accommodate distinct temporal behaviors. Time-domain modeling, like a Transformer, directly captures temporal dependencies underlying long-term trends, while frequency-domain modeling based on the Fourier transform provides explicit access to the amplitude and phase characteristics of fluctuations. This motivates spectrum-guided routing, which assigns decomposed modes according to their relative center frequencies: lower-frequency modes are directed to a time-domain branch to model long-term trends, while higher-frequency modes are routed to a frequency-domain branch to capture short-term fluctuations.

Based on the motivations described above, we propose Spectrum-DualNet for multivariate time series forecasting. Its Trend branch in the time domain employs a Transformer encoder with patches to represent gradual temporal evolution and long-range dependencies, while its Fluctuation branch in the frequency domain models amplitude-phase interactions (Liu et al., 2026) to characterize oscillatory dynamics. These complementary representations allow the model to capture temporal structure and spectral relationships within a single forecasting architecture. A shared prediction head maps the combined representation to future observations.

Our main contributions are summarized as follows:

- We introduce a spectrum-guided forecasting framework that couples multivariate mode decomposition with explicit time-frequency branch assignment, using relative mode frequencies to specialize the modeling of slow and oscillatory dynamics.
- We formulate a two-stage MVMD front end that calculates shared center frequencies from training data of different datasets and reuses them in a differentiable window-level filtering layer, giving the branches consistently ordered inputs.

- We design a dual-branch predictor with gated fusion of trend and fluctuation representations and a second gate for incorporating the decomposition residual, allowing their contributions to be learned from the forecasting objective.

2 Related Work

2.1 Time Series Decomposition

Time series decomposition separates entangled temporal dynamics into components to improve forecast efficiency. Autoformer incorporates moving-average decomposition into its network to separate trend and seasonal components (Wu et al., 2021), while DLinear applies separate linear predictors to the two components (Zeng et al., 2023). FEDformer extends this approach with a mixture of moving-average kernels and frequency-enhanced modeling (Zhou et al., 2022). Beyond a single temporal resolution, TimeMixer decomposes downsampled sequences into seasonal and trend components and mixes them across scales (Wang et al., 2024). TimeKAN constructs a moving-average hierarchy and uses cascaded frequency decomposition with frequency-specific KAN representations (Huang et al., 2025).

These methods demonstrate the value of separating temporal scales and tailoring their representations. However, they do not concentrate each component's spectral information around its data-estimated center frequencies. This spectrum information allows the decomposition to adapt to the characteristic timescales of the input, and yields spectral information that associates distinct frequency ranges with long-term and short-term patterns.

Adaptive mode decomposition provides a complementary approach. Empirical Mode Decomposition (EMD) extracts intrinsic mode functions according to local characteristic timescales (Huang et al., 1998). Variational Mode Decomposition (VMD) instead jointly estimates a prescribed number of modes and their center frequencies (Dragomiretskiy & Zosso, 2014), and Multivariate VMD (MVMD) extends this formulation to multiple channels (Rehman & Aftab, 2019). This shared-frequency formulation aligns corresponding modes across variables while allowing their amplitudes and phases to differ, providing components with consistent frequency interpretations for subsequent multivariate modeling. In Spectrum-DualNet, we use MVMD to estimate dataset-specific center frequencies from the training set and guide mode extraction and mode routing to the trend and fluctuation branches. Our two-stage design separates this frequency estimation from the sliding-window decomposition integrated in the model: the estimated centers are consistent and reused in a differentiable sequential filtering pass for each input window.

2.2 Frequency-Domain Forecasting

Frequency-domain forecasting represents temporal signals through spectral coefficients, providing a global view of periodic structure and an alternative view for learning temporal dependencies. FEDformer combines seasonal-trend decomposition with Fourier- and wavelet-based modules (Zhou et al., 2022). FreTS applies frequency-domain MLPs to the real and imaginary parts of spectral representations (Yi et al., 2023). FITS uses a complex-valued linear layer to interpolate frequency representations, learning amplitude scaling and phase shifts (Xu et al., 2024).

Apart from the aforementioned methods that utilize all spectrum components, another kind of research examines how different spectral components contribute to prediction. FilterNet introduces learnable frequency filters that selectively retain or attenuate signal components, using a shared frequency kernel in its plain shaping filter and input-dependent filtering in its contextual shaping filter (Yi et al., 2024). Amplifier addresses the tendency of forecasting models to emphasize high-energy components by introducing energy amplification and recovery operations that promote learning from lower-energy components (Fei et al., 2025). Beyond spectral energy, APRNet models amplitude-phase interactions from temporal and channel-wise perspectives to characterize non-stationary dynamics (Liu et al., 2026). Differently, our proposed Spectrum-DualNet adapts the amplitude–phase interaction mechanism proposed by APRNet within a channel-independent Fluctuation branch to capture non-stationary oscillatory dynamics, and integrates the resulting representation with a separately modeled trend representation.

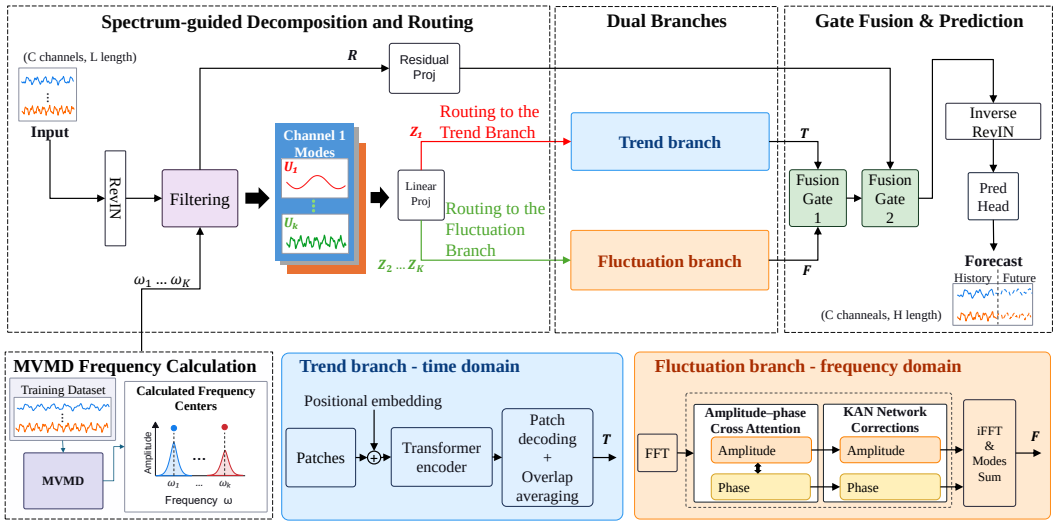


Figure 2: Overview of Spectrum-DualNet. The lower panels detail MVMD frequency estimation, the Trend branch, and the Fluctuation branch. Inputs are normalized by RevIN (Kim et al., 2021).

2.3 Multi-Branch Forecasting

Multi-branch forecasting distributes modeling across complementary components, temporal scales, or modalities and combines their representations or predictions. DLinear applies separate linear predictors to trend and seasonal components (Zeng et al., 2023), while TimeMixer aggregates predictions from multiple temporal resolutions through its Future-Multipredictor-Mixing module (Wang et al., 2024). Beyond separate parameterization, xPatch uses exponential seasonal-trend decomposition and assigns the resulting components to an MLP-based trend stream and a CNN-based seasonal stream (Stitsyuk & Choi, 2025). Complementary branches can also operate across modalities: TimeIG jointly models time series and their visual representations through a dual-branch architecture with cross-modal attention and dynamic weighted fusion (Cao et al., 2026).

These approaches illustrate how separate prediction pathways can accommodate heterogeneous temporal structure. However, these designs do not utilize the mode separation information to guide the choice of modeling mechanism. Meanwhile, frequency information from the decomposition process provides a basis for matching branch mechanisms to the dynamics of their inputs in Spectrum-DualNet. Spectrum-DualNet adapts the amplitude-phase interaction mechanism proposed by APRNet within a channel-independent Fluctuation branch to model non-stationary dynamics, then combines the resulting representation with a separately modeled trend representation.

3 Methodology

Spectrum-DualNet aims to disentangle temporal dynamics with different spectral components and model them with suitable forecasting mechanisms, as illustrated in Figure 2. To obtain data-adaptive components with consistent spectrum identities across channels and input windows, we use a two-stage decomposition that calculates shared center frequencies with MVMD and uses the frequencies for sliding-window filtering. To specialize modeling for slow variations and faster fluctuations, spectrum-guided routing assigns the resulting modes to a time-domain Trend branch and a frequency-domain Fluctuation branch. Finally, gated fusion adapts the contributions of the two branch representations to each input, while gated residual injection incorporates information not captured by the extracted modes before the final prediction.

3.1 Two-Stage Decomposition for Spectrum-Guided Routing

Spectrum-DualNet performs the decomposition in two stages. The first stage calculates dataset-specific spectral modes, which are defined by center frequencies shared across channels of the train-

ing sequence; the second reuses these frequencies to decompose each input window into modes in the time domain. Spectrum-guided routing then assigns the lowest-frequency mode, which captures relatively slow variations, to the time-domain Trend branch and the remaining higher-frequency modes to the frequency-domain Fluctuation branch. Such routing matches the modeling domain to the spectral characteristics of each group of modes.

Stage I: MVMD frequency calculation. We apply MVMD (Rehman & Aftab, 2019) to the multivariate training sequence, and assume that there are always modes to be decomposed ($K > 1$). MVMD jointly calculates the modes and their center frequencies by minimizing the aggregate mode bandwidth across channels under a reconstruction constraint; details are provided in Appendix A.1. Each mode has one center frequency shared across all channels. The estimated center frequencies, expressed in cycles per sample, are sorted as $0 \leq \omega_1 \leq \cdots \leq \omega_K \leq 1/2$ and retained for subsequent sliding-window processing. We also estimated the impact of the sliding-window strategy on the modes via experiments in section 4.3. Interestingly, the periods corresponding to the frequencies do have physical meanings; more details are provided in section 4.3. These frequencies are computed using only the training set and held consistent during training, validating, and testing. Thus, spectral adaptation occurs at the dataset level, and the calculations can be reused across forecasting runs for different prediction lengths with the same decomposition configuration.

Stage II: Spectrum-guided decomposition. This stage uses the center frequencies estimated in Stage I to extract modes from each input window. Let $\mathbf{X} \in \mathbb{R}^{C \times L}$ denote an input window and $\widetilde{\mathbf{X}}$ its RevIN-normalized counterpart, where C is the number of channels and L is the context length. We omit the batch dimension throughout and use $c \in \{1, \ldots, C\}$ to index channels. We extend the window to length $2L$ by repeating its boundary values, then apply a real Fourier transform to obtain its spectrum $\widehat{\mathbf{X}}(f)$, where the frequency axis $f \in \{m/(2L) : m = 0, \ldots, L\}$. Let $\widehat{\mathbf{U}}_j(f) \in \mathbb{C}^C$ denote the spectrum of the jth mode for the input window. We initialize all mode spectra to zero and extract the modes sequentially. At step k, the spectra $\widehat{\mathbf{U}}_1(f), \ldots, \widehat{\mathbf{U}}_{j-1}(f)$ have already been computed in the preceding steps. We subtract these extracted spectra from the input spectrum to obtain the remaining spectral content $\widehat{\mathbf{E}}_k(f)$:

$$\widehat{\mathbf{E}}_k(f) = \widehat{\mathbf{X}}(f) - \sum_{j=1}^{k-1} \widehat{\mathbf{U}}_j(f), \qquad k = 1, \ldots, K, \tag{1}$$

where the sum is zero for $k = 1$. The remaining spectrum is then filtered using the fixed center frequency ω_k:

$$H_k(f) = \frac{1}{1 + \alpha(f - \omega_k)^2}, \qquad \widehat{\mathbf{U}}_k(f) = H_k(f)\widehat{\mathbf{E}}_k(f), \tag{2}$$

The filter $H_k(f)$ has its largest response at ω_k and increasingly attenuates frequencies farther from this center, with larger α producing a narrower response. Thus, each mode extracts spectral content near its assigned center from the remaining spectrum. The same filter is applied to every channel, giving each mode a consistent frequency interpretation across variables while retaining channel-specific spectral values.

After extracting all K mode spectra, we transform them all back to the time domain and crop the padded regions to obtain $\mathbf{U}_k \in \mathbb{R}^{C \times L}$. Because this single filtering pass cannot assign all input content to the modes, we explicitly retain the reconstruction residual:

$$\mathbf{R} = \widetilde{\mathbf{X}} - \sum_{k=1}^{K} \mathbf{U}_k. \tag{3}$$

The residual is passed to the later residual gate so that information left outside the extracted modes remains available for forecasting. This stage performs one ordered filtering pass per window rather than repeating the iterative MVMD optimization. The centers and α remain consistent, while the filtering operations are differentiable with respect to the input.

Spectrum-guided routing. The ordered center frequencies provide a dataset-specific basis for assigning the decomposed modes to complementary forecasting mechanisms. The mode associated with the lowest center frequency, $\mathbf{U}_1$, represents relatively slow variations and is assigned to

the time-domain Trend branch, which models temporal dependencies through a patch-based Transformer. The remaining modes, $\{\mathbf{U}_k\}_{k=2}^{K}$, represent relatively faster oscillations and are assigned to the frequency-domain Fluctuation branch, which models their spectral amplitudes, phases, and interactions. Here, trend and fluctuation are defined by the relative ordering of the estimated center frequencies, without requiring a predefined frequency threshold.

Denoting the two branch mappings by $\mathcal{B}_{\text{trend}}$ and $\mathcal{B}_{\text{fluct}}$, spectrum-guided routing is expressed as

$$\mathbf{T} = \mathcal{B}_{\text{trend}}(\mathbf{Z}_1), \qquad (\mathbf{F}_2, \ldots, \mathbf{F}_K) = \mathcal{B}_{\text{fluct}}(\mathbf{Z}_2, \ldots, \mathbf{Z}_K), \tag{4}$$

where $\mathbf{T}, \mathbf{F}_k$ denote the trend representation and the reconstructed representation of each fluctuation mode respectively. This assignment couples the estimated spectral structure to branch specialization, producing complementary features $\mathbf{T}$ and $\mathbf{F}$ for subsequent gated fusion.

3.2 Time-Domain Trend Branch

The Trend branch processes the lowest-frequency mode $\mathbf{Z}_1 \in \mathbb{R}^{C\times D}$ using a patch Transformer encoder with an optional patching process proposed by Nie et al. (2023) to capture dependencies within its slowly varying representation. Each channel is processed independently with shared parameters. We divide the input into patches of length P and stride S, padding when necessary, and apply a linear embedding with learned positional embeddings. The patching process is set to be optional, and it is naturally disabled when both patch size P and stride S are set to be 1. A Transformer encoder models dependencies among the resulting tokens, which are then linearly decoded into patches. Overlapping outputs are averaged and padding is removed to obtain the trend representation $\mathbf{T} \in \mathbb{R}^{C\times D}$ for gated fusion. Setting $P = S = 1$ recovers point-wise tokenization for the Transformer.

3.3 Frequency-Domain Fluctuation Branch

The fluctuation branch models the higher-frequency modes through their spectral amplitudes and phases. The inputs are the projected modes $\mathbf{Z}_k \in \mathbb{R}^{C\times D}$, $k = 2, \ldots, K$. An orthonormal real Fourier transform is applied along the length-D dimension. When the learned projection is enabled, this transform describes the projected representation; the center frequencies ω_k used for routing remain consistent. We extract the magnitudes and phase angles of the modes and, for each channel c, stack the $K-1$ modes into amplitude and phase matrices $\mathbf{A}_c, \mathbf{\Phi}_c \in \mathbb{R}^{(K-1)\times F}$ where $F = \lfloor D/2 \rfloor + 1$ is the number of non-negative Fourier bins for length D. Row $k-1$ corresponds to mode k: $\mathbf{A}_c[k-1,:] = |\,\mathrm{FFT}_D(\mathbf{Z}_k[c,:])|$ and $\mathbf{\Phi}_c[k-1,:] = \arg(\mathrm{FFT}_D(\mathbf{Z}_k[c,:]))$, with phase measured in radians. These two representations describe the strength and alignment of the oscillatory components.

Amplitude-phase interactions. We use two cross-attention modules to exchange information between amplitude and phase across modes. For generic feature matrices $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{(K-1)\times F}$, each module computes

$$\mathrm{CrossAttn}(\mathbf{B}_1, \mathbf{B}_2) = \mathrm{LayerNorm}\left[\mathbf{B}_1 + \mathrm{softmax}\left(\frac{\mathcal{Q}(\mathbf{B}_1)\mathcal{K}(\mathbf{B}_2)^\top}{\sqrt{F}}\right)\mathcal{V}(\mathbf{B}_2)\right], \tag{5}$$

where $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ are learned affine maps from F to F features, applied to each row. Softmax operates over the source modes, and LayerNorm operates along the F spectral features. We obtain $\mathbf{M}_c^A = \mathrm{CrossAttn}_A(\mathbf{A}_c, \mathbf{\Phi}_c)$ and $\mathbf{M}_c^\Phi = \mathrm{CrossAttn}_\Phi(\mathbf{\Phi}_c, \mathbf{A}_c)$ using separate parameters. Each attention matrix has size $(K-1)\times(K-1)$: amplitude queries aggregate phase information from the fluctuation modes, and phase queries aggregate amplitude information. Channels are processed independently with shared module parameters.

Spectral correction. Separate KAN networks (Liu et al., 2025) g_A and g_Φ are implemented to transform features obtained above into amplitude gates and phase adjustments. Each network contains KAN layers followed by dropout and a linear projection to F features, acting independently on each mode row. The KAN layers combine learned Gaussian radial-basis expansions of normalized features with a SiLU-activated linear path. The resulting corrections are applied to the original spectra:

$$\mathbf{A}'_c = \mathbf{A}_c \odot \sigma\big(g_A(\mathbf{M}_c^A)\big), \qquad \mathbf{\Phi}'_c = \mathbf{\Phi}_c + \tanh\big(g_\Phi(\mathbf{M}_c^\Phi)\big), \tag{6}$$

where $\odot$ denotes element-wise multiplication and σ is the element-wise sigmoid function. Both corrected matrices remain in $\mathbb{R}^{(K-1)\times F}$. The sigmoid gate selectively attenuates amplitudes while preserving nonnegativity, and the phase adjustment is bounded between -1 and 1 radians. The mappings share parameters across modes and channels, while their outputs depend on the current input.

After spectral correction, we reconstruct each corrected complex spectrum and apply the inverse Fourier transform with output length D. Summing the reconstructed modes gives

$$\mathbf{F}_{c,:} = \sum_{k=2}^{K} \mathrm{iFFT}_D(\mathbf{A}'_c[k-1,:] \odot \exp(\mathrm{i}\mathbf{\Phi}'_c[k-1,:])), \tag{7}$$

where i is the imaginary unit and both transforms use orthonormal normalization. Stacking the channel outputs yields $\mathbf{F} \in \mathbb{R}^{C\times D}$ to match the trend representation for gated fusion.

3.4 Gated Fusion and Residual Incorporation

We first fuse the trend and fluctuation representations $\mathbf{T}, \mathbf{F} \in \mathbb{R}^{C\times D}$ through an input-dependent gate:

$$\mathbf{G}_1 = \sigma(g_1([\mathbf{F} \,\|\, \mathbf{T}])), \qquad \mathbf{H} = \mathbf{G}_1 \odot \mathbf{F} + (1 - \mathbf{G}_1) \odot \mathbf{T}, \tag{8}$$

where $\|$ denotes concatenation along the feature dimension, so $[\mathbf{F} \,\|\, \mathbf{T}] \in \mathbb{R}^{C\times 2D}$ and $\mathbf{G}_1, \mathbf{H} \in \mathbb{R}^{C\times D}$. Both gate mappings g_1 and g_2 are MLPs, mapping $2D$ features to D dimensions. The gates have independent parameters shared across channels and produce feature-wise mixing coefficients.

To incorporate information retained in the decomposition residual, we obtain $\mathbf{R}_D \in \mathbb{R}^{C\times D}$ through a separate affine projection from L to D, or use $\mathbf{R}_D = \mathbf{R}$ when projection is omitted. The second gate computes

$$\mathbf{G}_2 = \sigma(g_2([\mathbf{H} \,\|\, \mathbf{R}_D])), \qquad \mathbf{Z} = \mathbf{G}_2 \odot \mathbf{H} + (1 - \mathbf{G}_2) \odot \mathbf{R}_D. \tag{9}$$

Here $\mathbf{G}_2, \mathbf{Z} \in \mathbb{R}^{C\times D}$, and $1 - \mathbf{G}_i$ denotes the element-wise complement of gate i. A shared MLP-based prediction head maps each channel of $\mathbf{Z}$ from D features to H future steps, where H is the forecast horizon. Inverting RevIN with the input-window normalization statistics yields $\widehat{\mathbf{Y}} \in \mathbb{R}^{C\times H}$.

4 Experiments

Datasets Extensive experiments are conducted on six commonly used real-world datasets, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic. We used the popular TSLib from `https://github.com/thuml/Time-Series-Library` to split the datasets and generate the windows. All datasets are downloaded from Autoformer (Wu et al., 2021) as TSLib required.

Experimental Settings To ensure fair comparisons, we use a fixed input length of 512 for all datasets with prediction lengths $H \in \{96, 192, 336, 720\}$. We used the weight of L1 and L2 loss to train the model, and the weight is set to be a hyperparameter. Both MSE and MAE are used to evaluate the model performance. All experiments are conducted on an NVIDIA RTX A5000 GPU.

Baselines We selected eight representative long-term time series forecasting methods from published papers, including FreqCycle by Zhang et al. (2026), APRNet by Liu et al. (2026), TimeKAN by Huang et al. (2025), Amplifier by Fei et al. (2025), iTransformer by Liu et al. (2024), FilterNet by Yi et al. (2024), PatchTST by Nie et al. (2023), and DLinear by Zeng et al. (2023). We reference the results of the models that have the same experimental settings from the papers. For models that have different experimental settings, we reproduce the model under our experimental settings.

4.1 Main Results

Table 1 reports the long-term forecasting results of Spectrum-DualNet and eight baseline methods, averaged over four forecasting horizons. The full results are provided in Appendix A.2. Across seven datasets and four forecasting horizons, Spectrum-DualNet achieves the best or the second best MSE in 28 of the 35 settings and the best or second best MAE in 30 settings. When results are averaged over the four horizons, Spectrum-DualNet achieves the best or second best MSE and MAE on six datasets among the seven datasets.

Table 1: Forecasting performance on seven datasets, averaged over prediction lengths 96, 192, 336, and 720. The best and second-best results are bolded and underlined, respectively. Full results available in Appendix A.2

Dataset	Spectrum-DualNet		FreqCycle (2026)		APRNet (2026)		TimeKAN (2025)		Amplifier (2025)		iTransformer (2024)		FilterNet (2024)		PatchTST (2023)		DLinear (2023)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.403**	0.422	0.414	0.435	0.411	0.427	0.404	**0.421**	0.428	0.443	0.451	0.462	0.416	0.432	0.413	0.434	0.422	0.437
ETTh2	0.352	0.389	0.357	0.401	0.336	0.384	0.357	0.402	0.335	0.393	0.382	0.414	0.378	0.416	**0.331**	**0.381**	0.431	0.446
ETTm1	**0.347**	**0.373**	0.356	0.383	0.348	0.378	0.352	0.385	0.377	0.392	0.370	0.399	0.357	0.386	0.353	0.382	0.357	0.378
ETTm2	**0.253**	**0.312**	0.256	0.318	**0.253**	0.314	0.261	0.322	0.265	0.325	0.272	0.331	0.268	0.329	0.256	0.317	0.267	0.333
Weather	**0.220**	**0.253**	0.228	0.267	0.223	0.262	0.221	0.262	0.262	0.286	0.305	0.335	0.229	0.271	0.226	0.264	0.248	0.300
Electricity	0.159	**0.249**	0.161	0.255	**0.155**	0.251	0.163	0.356	0.168	0.265	0.203	0.298	0.162	0.257	0.159	0.253	0.166	0.263
Traffic	**0.388**	0.261	0.398	**0.243**	**0.388**	0.276	0.412	0.280	0.404	0.282	0.397	0.295	0.395	0.280	0.391	0.264	0.433	0.295

Table 2: Comparison of the full Spectrum-DualNet model with variants that individually remove decomposition, the Trend branch, or the Fluctuation branch to assess the benefit of each component across seven datasets. Lower MSE and MAE indicate better performance; the best results are bolded.

Dataset	Full model		w/o Decomposition		w/o Trend Branch		w/o Fluctuation Branch	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.360**	**0.391**	0.368	0.399	0.385	0.414	0.415	0.438
ETTh2	**0.269**	**0.330**	0.286	0.346	0.331	0.389	0.294	0.350
ETTm1	**0.284**	**0.338**	0.287	0.341	0.305	0.365	0.292	0.346
ETTm2	**0.162**	**0.251**	0.165	0.253	0.244	0.316	0.164	0.252
Weather	**0.143**	**0.187**	0.152	0.200	0.144	0.190	0.149	0.200
Electricity	0.127	0.217	0.130	0.223	**0.124**	**0.216**	0.131	0.225
Traffic	**0.354**	**0.239**	0.365	0.251	0.370	0.258	0.369	0.357

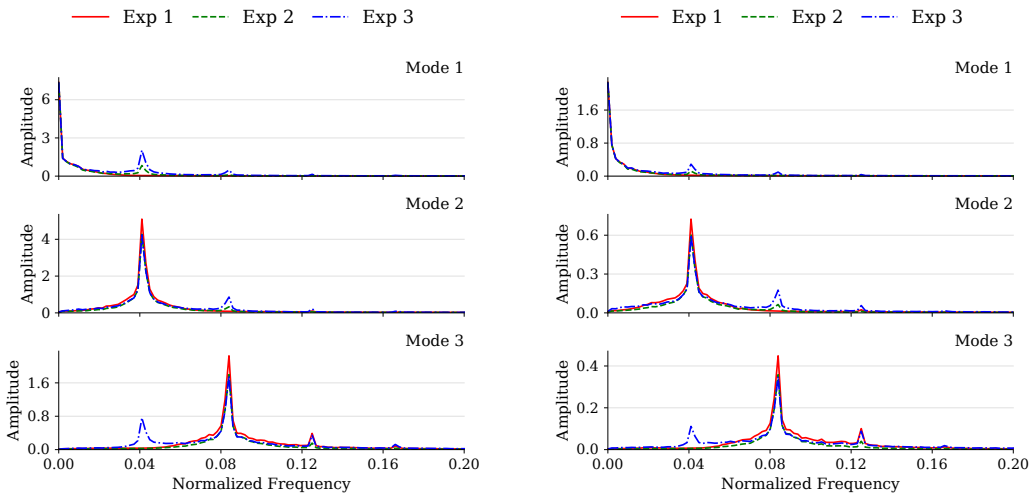
4.2 Ablation Study

Table 2 compares the full Spectrum-DualNet model with variants that individually remove the decomposition module, the Trend Branch, or the Fluctuation Branch. The decomposition module is disabled by sending the input to the branches directly; the branch is ablated by replacing the input of Fusion Gate 2 with the output of another branch, while keeping the residual. The ablation study is conducted under the same experimental settings as the previous Main Results section. The results indicate that all the three components are essential for the model.

4.3 Mode Decomposition

In the decomposition stage, Spectrum-DualNet uses center frequencies obtained from the whole training set to decompose the data split into windows. To estimate the impact of this inconsistency, we conducted three experiments on all the datasets: Exp 1 applies classical MVMD to the entire dataset, including the validation and test sets; Exp 2 only uses the training set to calculate the center frequencies; Exp 3 uses the center frequencies from Exp 2 and processes the dataset window by window to match the data form required by time series models. The experimental results indicate that windowing generally has a limited impact on the spectrum, and that such impact can be absorbed by the downstream model structures. Figure 3 displays the experimental results for Channels 0 and 1 of the ETTh1 dataset.

Table 3 summarizes the center frequencies obtained through mode decomposition and the periods calculated using the frequencies and the corresponding sampling rates for electricity-related datasets. All datasets, except ETTm datasets, are sampled hourly. The sampling rate of ETTm datasets is 15



(a) Channel 0 (b) Channel 1

Figure 3: Effect of training-only frequency estimation and sliding-window filtering on the extracted mode spectra for Channels 0 and 1 of ETTh1. We compare classical MVMD applied to the entire dataset (Exp 1), classical MVMD applied only to the training set (Exp 2), and sliding-window filtering using the fixed center frequencies estimated in Exp 2 (Exp 3, ours). The results indicate that training-only frequency estimation and sliding-window filtering largely preserve the dominant spectral structure of the extracted modes, with limited changes in their spectral profiles.

Table 3: Dataset-specific center frequencies estimated by MVMD and the calculated periods using the frequencies. Most datasets approximately exhibit monthly, daily, and half-daily periodic patterns, consistent with periodicities of electricity demand (Luque et al., 2020).

Dataset	Mode 1		Mode 2		Mode 3	
	Frequency	Period	Frequency	Period	Frequency	Period
ETTh1	0.0012	34.7 days	0.0412	24.2 hours	0.0845	11.8 hours
ETTh2	0.0015	27.8 days	0.0402	24.9 hours	-	-
ETTm1	0.0004	24.1 days	0.0119	21.0 hours	0.0411	6.1 hours
ETTm2	0.0010	10.4 days	0.0732	3.4 hours	-	-
Electricity	0.0021	19.8 days	0.0416	24.0 hours	0.0847	11.8 hours

minutes per sample. The calculated periodic patterns are consistent with periodicities reported in spectral analysis of electricity demand by Luque et al. (2020), and that suggests the decomposition method used in Spectrum-DualNet can effectively extract the periodic components at multiple timescales.

5 Conclusion

We presented Spectrum-DualNet, a multivariate time series forecasting framework that combines adaptive mode decomposition with spectrum-guided routing to complementary time- and frequency-domain branches. Its two-stage decomposition estimates dataset-specific center frequencies from training data and reuses them for window-level filtering, while gated fusion integrates trend, fluctuation, and residual information. Experiments on seven benchmark datasets demonstrate competitive forecasting performance, and ablation results support the benefits of decomposition and branch specialization across datasets. These findings highlight the potential of matching forecast mechanisms to the spectral characteristic of temporal components.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, assist in the writing of proofs, implement methods, and assist with translation.

We have not used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, interpret results, and generate synthetic data sets, provide critical ingredients for proving mathematical claims, clean and reformat datasets.

Additionally, we used generative AI tools to create or edit software code, summarize or analyse existing literature, draft parts of a research paper, brainstorm, and identify relevant literature. We have reviewed all AI-assisted work. Modifications made by an LLM in the code of the paper were verified and tested for correctness. All content in the paper was also checked to ensure our ideas are expressed properly. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

After the discussion opens, the code will be available to the reviewers via a comment containing a link to an anonymous repository. All code will be publicly available in a GitHub repository to facilitate replication and verification once the paper is accepted. The experimental setup, including training steps, model configurations, and hardware details, is described in the paper, and the corresponding scripts will be available alongside the code. We have also provided a full description of our contribution, to assist others in reproducing our experiments. Additionally, all the datasets used in the paper are publicly available, ensuring consistent and reproducible evaluation results.

REFERENCES

Minjun Cao, Hao Miao, Wentao Zhang, and Senzhang Wang. From diversity to uniformity: Cross-modal time series modeling with dependent channel grouping. In Diego Calvanese (ed.), *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pp. 2447–2455. International Joint Conferences on Artificial Intelligence Organization, 8 2026. doi: 10.24963/ijcai.2026/272. URL https://doi.org/10.24963/ijcai.2026/272. Main Track.

Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE Transactions on Signal Processing*, 62(3):531–544, 2014. doi: 10.1109/TSP.2013.2288675.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11645–11653, 2025. doi: 10.1609/aaai.v39i11.33267.

Norden E. Huang, Zheng Shen, Steven R. Long, Manli C. Wu, Hsing H. Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971):903–995, 1998. doi: 10.1098/rspa.1998.0193.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=wTLc79YNbh.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges. *Artificial Intelligence Review*, 58:216, 2025. doi: 10.1007/s10462-025-11223-9.

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

International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Hao Liu, Xiaobin Zhu, Chun Yang, Lirui Deng, and Xucheng Yin. Amplitude–phase reconstruction for non-stationary time-series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 38(8):4868–4882, 2026. doi: 10.1109/TKDE.2026.3701273.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 70367–70413, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/afaed89642ea100935e39d39a4da602c-Paper-Conference.pdf.

Joaquin Luque, Davide Anguita, Francisco Pérez, and Robert Denda. Spectral analysis of electricity demand using hilbert–huang transform. *Sensors*, 20(10):2912, 2020. doi: 10.3390/s20102912.

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

Naveed ur Rehman and Hania Aftab. Multivariate variational mode decomposition. *IEEE Transactions on Signal Processing*, 67(23):6039–6052, 2019. doi: 10.1109/TSP.2019.2951223.

Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, 1st edition, 1997.

Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

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*, volume 34, pp. 22419–22430, 2021.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/701251e1db4a2e4dd2ef23f5265d5936-Abstract-Conference.html.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain MLPs are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. FilterNet: Harnessing frequency filters for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6323d96f79d5d49e0d3fc88835c082cd-Abstract-Conference.html.

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*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. Freqcycle: A multi-scale time-frequency analysis method for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28159–28167, 2026. doi: 10.1609/aaai.v40i33.40042.

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 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Appendix

A.1 Multivariate Variational Mode Decomposition

This appendix gives the complete iterative procedure used to estimate the shared center frequencies in Stage I of Section 3.1. Following MVMD (Rehman & Aftab, 2019), each multivariate mode has a common spectral center across channels, while its amplitude and phase can differ between channels. We present the discrete frequency-domain formulation with the normalization used in our implementation.

Input and spectral representation. Let $\mathbf{X}^{\mathrm{tr}} \in \mathbb{R}^{C\times L_{\mathrm{tr}}}$ denote the preprocessed training sequence, with C channels and L_{tr} observations. The superscript tr distinguishes full-training-sequence quantities from the window-level input $\mathbf{X}$ and modes $\mathbf{U}_k$ in Section 3.1; L_{tr} is distinct from the context length L and feature dimension D. We retain the main-text indices $c = 1,\dots,C$ and $k = 1,\dots,K$, with integer $K \geq 2$. Only the training split is used for frequency estimation. To reduce boundary discontinuities, symmetrically extend each channel by $\lfloor L_{\mathrm{tr}}/2 \rfloor$ samples on the left and $L_{\mathrm{tr}} - \lfloor L_{\mathrm{tr}}/2 \rfloor$ samples on the right. Denote the resulting length-$2L_{\mathrm{tr}}$ sequence by $\mathbf{X}^{\mathrm{tr,ext}}$. Its nonnegative-frequency Fourier coefficients are

$$\widehat{X}_c^{\mathrm{tr}}(f_m) = \sum_{t=0}^{2L_{\mathrm{tr}}-1} X_{c,t}^{\mathrm{tr,ext}} \exp(-\mathrm{i}2\pi f_m t), \qquad f_m = \frac{m}{2L_{\mathrm{tr}}}, \quad m = 0,\dots,L_{\mathrm{tr}}. \tag{10}$$

Here t indexes padded time samples, m indexes nonnegative-frequency bins, and $\mathrm{i}^2 = -1$. The grid $f_m = m/(2L_{\mathrm{tr}})$ is the training-sequence counterpart of the window grid $f = m/(2L)$ in Stage II; both use cycles per sample and have Nyquist frequency $1/2$. Let $\mathbf{U}_k^{\mathrm{tr}} \in \mathbb{R}^{C\times L_{\mathrm{tr}}}$ denote the unpadded training-sequence mode. Its padded spectrum is $\widehat{\mathbf{U}}_k^{\mathrm{tr}}(f_m) \in \mathbb{C}^C$, with scalar channel entries $\widehat{U}_{k,c}^{\mathrm{tr}}(f_m)$; similarly, $\widehat{X}_c^{\mathrm{tr}}(f_m)$ is the cth entry of $\widehat{\mathbf{X}}^{\mathrm{tr}}(f_m)$. Each mode has a shared center $\omega_k \in [0, 1/2]$, estimated here and sorted at termination to obtain the centers used in the main text.

Variational objective. MVMD measures bandwidth through the temporal derivative of each analytic mode after demodulation to its center frequency. In the Fourier domain, this corresponds to penalizing spectral energy by its squared distance from that center. With constant Fourier and analytic-signal scaling factors absorbed into the bandwidth coefficient, the discrete objective is

$$\begin{aligned} \min_{\{\widehat{U}_{k,c}^{\mathrm{tr}}\},\{\omega_k\}} \quad & \sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{m=0}^{L_{\mathrm{tr}}} (f_m - \omega_k)^2 |\widehat{U}_{k,c}^{\mathrm{tr}}(f_m)|^2, \\ \text{subject to} \quad & \sum_{k=1}^{K} \widehat{U}_{k,c}^{\mathrm{tr}}(f_m) = \widehat{X}_c^{\mathrm{tr}}(f_m), \qquad \forall c, m. \end{aligned} \tag{11}$$

The common ω_k couples the channels when estimating each mode's spectral location. It does not require equal channel spectra. Define the spectral constraint violation as $\rho_c(f_m) = \sum_{k=1}^{K} \widehat{U}_{k,c}^{\mathrm{tr}}(f_m) - \widehat{X}_c^{\mathrm{tr}}(f_m)$. This quantity is defined as the sum of mode spectra minus the input spectrum; it is distinct from the window-level time-domain residual $\mathbf{R}$, whose sign convention is input minus modes.

Using complex multipliers $\lambda_c(f_m)$, the augmented objective is

$$\mathcal{L} = \alpha \sum_{k,c,m} (f_m - \omega_k)^2 |\widehat{U}^{\mathrm{tr}}_{k,c}(f_m)|^2 + \sum_{c,m} |\rho_c(f_m)|^2 + \mathrm{Re} \sum_{c,m} \lambda_c(f_m)^* \rho_c(f_m), \tag{12}$$

where $\alpha > 0$ is the same bandwidth coefficient used in the Stage II filter $H_k(f)$, Re denotes the real part, and * denotes complex conjugation. The abbreviated sums run over $k = 1, \ldots, K$, $c = 1, \ldots, C$, and $m = 0, \ldots, L_{\mathrm{tr}}$ as applicable. This convention gives the denominator $1+\alpha(f_m-\omega_k)^2$ used in both decomposition stages.

Initialization and alternating updates. Let $n = 0, 1, \ldots$ index optimization iterations, indicated by parenthesized superscripts. Initialize $\widehat{U}^{\mathrm{tr},(0)}_{k,c}(f_m) = 0$ and $\lambda^{(0)}_c(f_m) = 0$ for all k, c, m. The default initialization sets $\omega^{(0)}_k = 0$ for every k; the implementation also supports uniformly spaced centers on $[0, 1/2]$ and logarithmically spaced centers from 0.0005 to 0.5. The centers are subsequently updated, including the lowest one; no mode is constrained to remain at zero frequency. At iteration n, perform the mode and center updates for each k in sequence, followed by one multiplier update after all K modes have been processed.

1. **Update the mode spectra.** For $k = 1, \ldots, K$, minimize Equation (12) with respect to the current mode while holding the other quantities fixed:

$$\widehat{U}^{\mathrm{tr},(n+1)}_{k,c}(f_m) = \frac{\widehat{X}^{\mathrm{tr}}_c(f_m) - \sum_{j=1}^{k-1} \widehat{U}^{\mathrm{tr},(n+1)}_{j,c}(f_m) - \sum_{j=k+1}^{K} \widehat{U}^{\mathrm{tr},(n)}_{j,c}(f_m) - \lambda^{(n)}_c(f_m)/2}{1 + \alpha(f_m - \omega^{(n)}_k)^2}. \tag{13}$$

   This update is applied to every channel and frequency bin, with empty sums interpreted as zero. The sequential sweep uses the newly updated spectra for modes $j < k$ and the preceding iteration's spectra for modes $j > k$.

2. **Update the shared centers.** After updating mode k, calculate its energy-weighted frequency centroid over all channels:

$$\omega^{(n+1)}_k = \frac{\sum_{c=1}^{C} \sum_{m=0}^{L_{\mathrm{tr}}} f_m |\widehat{U}^{\mathrm{tr},(n+1)}_{k,c}(f_m)|^2}{\sum_{c=1}^{C} \sum_{m=0}^{L_{\mathrm{tr}}} |\widehat{U}^{\mathrm{tr},(n+1)}_{k,c}(f_m)|^2}. \tag{14}$$

   Pooling energy across channels produces one center per multivariate mode. For a nonzero mode, the centroid lies in $[0, 1/2]$. The quotient requires positive total mode energy; a zero-energy mode has no identifiable center and requires a numerical safeguard, such as retaining its previous center.

3. **Update the multipliers.** After all K modes have been updated, take a dual-ascent step with step size $\tau > 0$:

$$\lambda^{(n+1)}_c(f_m) = \lambda^{(n)}_c(f_m) + \tau \left(\sum_{k=1}^{K} \widehat{U}^{\mathrm{tr},(n+1)}_{k,c}(f_m) - \widehat{X}^{\mathrm{tr}}_c(f_m) \right). \tag{15}$$

   The sign of this update follows the definition of ρ_c and is consistent with the negative multiplier term in Equation (13).

Stopping criterion and reconstruction. After each complete sweep, evaluate the change in the mode spectra:

$$\Delta^{(n+1)} = \frac{1}{L_{\mathrm{tr}}} \sum_{k=1}^{K} \sum_{c=1}^{C} \sum_{m=0}^{L_{\mathrm{tr}}} \left| \widehat{U}^{\mathrm{tr},(n+1)}_{k,c}(f_m) - \widehat{U}^{\mathrm{tr},(n)}_{k,c}(f_m) \right|^2. \tag{16}$$

Here $\Delta^{(n+1)}$ is the squared spectral change normalized by the unpadded training length. Stop when $\Delta^{(n+1)} \leq \varepsilon$, where $\varepsilon > 0$ is the convergence tolerance, or the iteration count reaches the maximum $N_{\max}$. The forecasting entry point defaults to $\varepsilon = 10^{-4}$, $\tau = 10^{-2}$, $N_{\max} = 500$, $K = 3$, and $\alpha = 4000$; K and α specify the decomposition configuration. This stopping rule measures iterate change, so finite termination does not by itself guarantee an exactly zero reconstruction error or a global optimum.

Let n_* denote the number of completed iterations. At this final iteration, complete each mode's spectrum by Hermitian symmetry, apply the inverse Fourier transform, and remove the padding to recover $\mathbf{U}_k^{\mathrm{tr}} \in \mathbb{R}^{C \times L_{\mathrm{tr}}}$. Let π be a permutation of $\{1, \ldots, K\}$ that sorts the final centers, so that $\omega_{\pi(1)}^{(n_*)} \leq \cdots \leq \omega_{\pi(K)}^{(n_*)}$. Define the main-text centers by $\omega_k = \omega_{\pi(k)}^{(n_*)}$ and apply the same permutation to the reconstructed modes and their spectra. The outputs are the reconstructed modes, their spectra, and the ordered centers $0 \leq \omega_1 \leq \cdots \leq \omega_K \leq 1/2$; the center-frequency trajectory may also be retained for inspection.

Use in Spectrum-DualNet. The two-stage decomposition in Section 3.1 directly builds on the MVMD updates described above. Stage I applies Equations (13)–(15) iteratively to the training sequence and retains the final sorted center frequencies $\{\omega_k\}_{k=1}^{K}$. These centers are shared across channels and remain fixed, together with the bandwidth coefficient α, during window-level decomposition.

Stage II obtains Equations (1) and (2) by simplifying the MVMD mode update in Equation (13). Specifically, we replace the training-sequence spectrum with the padded, normalized input-window spectrum, initialize all window-level mode spectra and multipliers to zero, and perform a single sequential sweep with the Stage I centers held fixed. At step k, the previously extracted modes contribute the term $\sum_{j<k} \widehat{\mathbf{U}}_j(f)$, while the unprocessed modes and the multiplier term are zero. The numerator therefore becomes the remaining spectrum $\widehat{\mathbf{E}}_k(f)$ in Equation (1), and the reciprocal of the MVMD denominator gives the filter $H_k(f) = 1/[1 + \alpha(f - \omega_k)^2]$ in Equation (2). No center-frequency or multiplier updates are performed, and the sweep is not repeated.

This construction preserves the frequency-selective form of the MVMD mode update but does not solve the constrained MVMD problem for each window. Consequently, the extracted modes need not fully reconstruct the input; Equation (3) retains the unassigned content as a residual, which enters the forecasting network through a learned gate. The filtering operations are differentiable with respect to the input, while the centers and α remain fixed. The forecasting loss trains the branches, fusion gates, and prediction head to use these spectral components and the residual for prediction.

A.2 FULL FORECAST RESULTS

Table 4: Forecasting performance on seven datasets with prediction lengths of 96, 192, 336, and 720. Avg. denotes the average over the four prediction lengths. Lower MSE and MAE indicate better performance. The best and the second-best results are bolded and underlined, respectively.

Dataset	Horizon	Spectrum-DualNet		FreqCycle (2026)		APRNet (2026)		TimeKAN (2025)		Amplifier (2025)		iTransformer (2024)		FilterNet (2024)		PatchTST (2023)		DLinear (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.360**	**0.391**	0.373	0.403	0.368	0.395	**0.360**	0.394	0.372	0.401	0.395	0.420	0.381	0.408	0.370	0.400	0.375	0.399
	192	0.402	0.417	0.404	0.425	0.405	0.417	**0.397**	0.415	**0.397**	**0.414**	0.427	0.441	0.412	0.427	0.413	0.429	0.405	0.416
	336	**0.416**	0.426	0.434	0.452	0.427	0.430	**0.416**	**0.424**	0.434	0.450	0.445	0.457	0.431	0.439	0.422	0.440	0.439	0.443
	720	**0.434**	0.454	0.443	0.461	0.444	0.463	0.442	**0.452**	0.510	0.508	0.537	0.530	0.438	0.454	0.447	0.468	0.472	0.490
	Avg.	**0.403**	0.422	0.414	0.435	0.411	0.427	0.404	**0.421**	0.428	0.443	0.451	0.462	0.416	0.432	0.413	0.434	0.422	0.437
ETTh2	96	**0.269**	**0.330**	0.285	0.349	0.270	0.336	0.280	0.342	0.270	0.343	0.304	0.360	0.300	0.360	0.274	0.337	0.289	0.353
	192	0.355	0.383	0.353	0.394	0.330	**0.374**	0.335	0.385	0.323	0.379	0.377	0.403	0.369	0.404	0.341	0.382	**0.283**	0.418
	336	0.378	0.405	0.376	0.414	0.356	0.399	0.378	0.419	0.354	0.415	0.405	0.429	0.389	0.428	**0.329**	**0.384**	0.448	0.465
	720	0.405	0.437	0.416	0.447	0.386	0.427	0.433	0.461	0.394	0.436	0.443	0.464	0.454	0.471	**0.379**	**0.422**	0.605	0.551
	Avg.	0.352	0.389	0.357	0.401	0.336	0.384	0.357	0.402	0.335	0.393	0.382	0.414	0.378	0.416	**0.331**	**0.381**	0.431	0.446
ETTm1	96	**0.284**	**0.338**	0.298	0.351	0.288	0.344	0.298	0.354	0.315	0.356	0.312	0.366	0.318	0.358	0.293	0.346	0.299	0.343
	192	**0.326**	**0.361**	0.336	0.374	0.330	0.366	0.329	0.370	0.348	0.377	0.347	0.385	0.338	0.378	0.333	0.370	0.335	0.365
	336	**0.360**	**0.382**	0.368	0.391	0.361	0.386	0.361	0.390	0.383	0.396	0.379	0.404	0.368	0.395	0.369	0.392	0.369	0.386
	720	0.416	**0.412**	0.422	0.417	0.415	0.418	0.420	0.424	0.460	0.440	0.441	0.442	**0.403**	0.413	0.416	0.420	0.425	0.421
	Avg.	**0.347**	**0.373**	0.356	0.383	0.348	0.378	0.352	0.385	0.377	0.392	0.370	0.399	0.357	0.386	0.353	0.382	0.357	0.378
ETTm2	96	**0.162**	**0.251**	0.172	0.260	0.164	0.256	0.167	0.259	0.172	0.263	0.179	0.271	0.174	0.257	0.166	0.256	0.167	0.269
	192	**0.220**	**0.290**	0.224	0.299	0.221	0.293	0.223	0.297	0.230	0.300	0.242	0.313	0.240	0.330	0.223	0.296	0.224	0.303
	336	**0.271**	**0.325**	0.273	0.329	**0.271**	**0.325**	0.280	0.335	0.292	0.343	0.288	0.344	0.297	0.339	0.274	0.329	0.281	0.342
	720	0.358	**0.382**	0.356	0.383	**0.355**	0.384	0.374	0.395	0.366	0.394	0.378	0.397	0.362	0.389	0.362	0.385	0.397	0.421
	Avg.	**0.253**	**0.312**	0.256	0.318	**0.253**	0.314	0.261	0.322	0.265	0.325	0.272	0.331	0.268	0.329	0.256	0.317	0.267	0.333
Weather	96	**0.143**	**0.187**	0.148	0.201	0.146	0.197	0.144	0.197	0.163	0.218	0.253	0.304	0.151	0.204	0.149	0.198	0.176	0.237
	192	**0.187**	**0.230**	0.190	0.241	0.190	0.240	**0.187**	0.238	0.203	0.251	0.280	0.319	0.198	0.248	0.194	0.241	0.220	0.282
	336	**0.236**	**0.270**	0.245	0.283	0.240	0.279	0.238	0.279	0.284	0.306	0.321	0.344	0.247	0.285	0.245	0.282	0.265	0.319
	720	**0.313**	**0.324**	0.331	0.343	0.314	0.331	**0.313**	0.334	0.396	0.370	0.364	0.374	0.318	0.345	0.314	0.334	0.333	0.362
	Avg.	**0.220**	**0.253**	0.228	0.267	0.223	0.262	0.221	0.262	0.262	0.286	0.305	0.335	0.229	0.271	0.226	0.264	0.248	0.300
Electricity	96	0.127	**0.217**	0.130	0.225	**0.125**	0.221	0.137	0.340	0.138	0.235	0.147	0.248	0.127	0.224	0.129	0.222	0.140	0.237
	192	**0.147**	**0.237**	**0.147**	0.241	0.150	0.246	0.148	0.384	0.159	0.257	0.165	0.267	0.150	0.244	**0.147**	0.240	0.153	0.249
	336	0.162	**0.254**	0.164	0.260	**0.160**	0.257	0.165	0.406	0.177	0.273	0.178	0.279	0.171	0.267	0.163	0.259	0.169	0.267
	720	0.198	0.286	0.203	0.295	**0.185**	**0.279**	0.203	0.295	0.199	0.293	0.322	0.398	0.200	0.293	0.197	0.290	0.203	0.301
	Avg.	0.159	**0.249**	0.161	0.255	**0.155**	0.251	0.163	0.356	0.168	0.265	0.203	0.298	0.162	0.257	0.159	0.253	0.166	0.263
Traffic	96	0.354	0.239	0.372	**0.232**	**0.352**	0.257	0.386	0.269	0.387	0.272	0.367	0.288	0.366	0.266	0.360	0.249	0.410	0.282
	192	0.378	0.255	0.379	**0.236**	**0.376**	0.269	0.400	0.273	0.388	0.271	0.378	0.293	0.384	0.275	0.379	0.256	0.423	0.287
	336	**0.392**	0.264	0.395	**0.243**	**0.392**	0.277	0.411	0.279	0.407	0.286	0.399	0.294	0.395	0.280	**0.392**	0.264	0.436	0.296
	720	**0.427**	0.284	0.447	**0.260**	0.433	0.299	0.450	0.300	0.435	0.298	0.442	0.304	0.433	0.299	0.432	0.286	0.466	0.315
	Avg.	**0.388**	0.261	0.398	**0.243**	**0.388**	0.276	0.412	0.280	0.404	0.282	0.397	0.295	0.395	0.280	0.391	0.264	0.433	0.295