DSR-TSF: Spectrum-Driven Dynamic Routing for Efficient Long-Horizon Time Series Forecasting

Hua Wang
School of Computer and Artificial Intelligence
Ludong University
Yantai 264025, China
hwa229@163.com

Bing Li
School of Computer and Artificial Intelligence
Ludong University
Yantai 264025, China
libing@ldu.edu.cn

Fan Zhang*
School of Computer Science and Technology
Shandong Technology and Business University
Yantai 264005, China
zhangfan@sdtbu.edu.cn

Abstract

Long-horizon time series forecasting requires a careful balance between predictive accuracy and computational cost. Segmented temporal modeling has therefore become a practical strategy for reducing the burden of long-range prediction. However, real-world time series often contain superposed multi-periodic components, whose spectral structures vary across time, variables, and samples. Such variability limits fixed aggregation or frequency-modeling mechanisms, making them less adaptive to non-stationary periodic patterns and potentially causing informative components to be attenuated over long forecasting horizons. To address this limitation, we propose DSR-TSF, a spectrum-driven dynamic routing framework for long-horizon forecasting. DSR-TSF adaptively selects and combines multiple period-specific branches conditioned on the spectral characteristics of the input sequence, thereby improving the model's ability to capture non-stationary periodic structures. It further learns period-aware dynamic weights to modulate the contributions of multi-periodic patterns while retaining the segmented prediction structure. Without complex resampling or explicit period alignment, DSR-TSF provides a compact mechanism for modeling spectral heterogeneity and achieves competitive performance across seven datasets with diverse dimensionalities and temporal characteristics. **Code Repository:** https://github.com/leyiice/DSR-TSF

1 Introduction

Time series forecasting [Xiao and Liu, 2025] is fundamental to transportation, energy, meteorology, and industrial monitoring, yet long-horizon prediction remains challenging [Wang et al., 2026c]. As the horizon increases, models must capture long-range dependencies, non-stationary variations [Xiao and Hou, 2026], and periodic patterns while keeping computation manageable [Liu et al., 2025a, Wu et al., 2021].

Existing methods can be broadly grouped into representation-intensive and lightweight paradigms, with recent analyses motivating simpler alternatives to increasingly complex Transformer variants [Chen et al., 2025]. The former strengthens temporal modeling through patch-based, frequency-

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

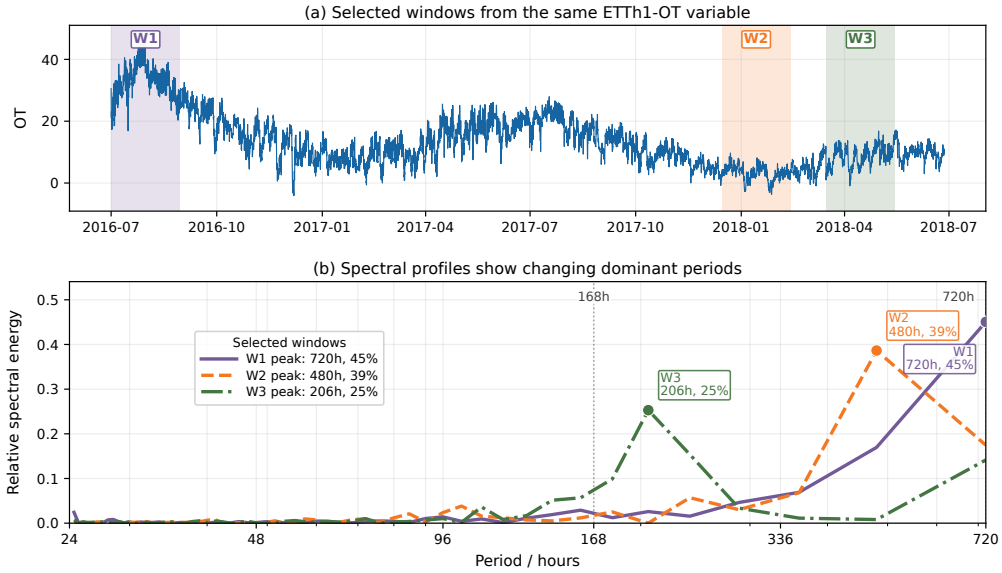


Figure 1: Dynamic spectral states across input windows.

domain, MLP-based, or multi-periodic representations [Nie et al., 2022, Yi et al., 2023, Wu et al., 2023, Tang and Zhang, 2025], often at higher computational cost. The latter, including DLinear [Zeng et al., 2023], SOFTS [Han et al., 2024], and VarDrop [Kang et al., 2025], improves efficiency through compact architectures, reduced representations, or lower token usage. However, many lightweight designs still use fixed aggregation or prediction schemes, making them less responsive to spectral variations across windows, variables, or samples. Improving adaptability to changing spectral states while retaining compact segmented forecasting therefore remains insufficiently explored.

As shown in Fig. 1, the dominant periods of the same ETTh1 target variable shift across windows, from 720h to 480h and 206h. This variation suggests that a shared segmented aggregation path may become mismatched with the current periodic state, weakening long-horizon modeling. Although frequency-domain learning, cross-frequency modeling, and non-stationarity adaptation have been studied [Yi et al., 2023, Kou et al., 2025, Lu et al., 2026], using spectral cues to condition compact segmented forecasting remains underexplored.

To address this issue, we propose DSR-TSF, a period-aware spectrum-driven dynamic routing framework for long-horizon time series forecasting. Instead of applying a uniform periodic modeling strategy to all inputs, DSR-TSF extracts compact spectral descriptors from the input sequence and maps them into routing weights that adaptively combine multiple period-response branches. This design allows the aggregation process to better adapt to the current spectral state without requiring explicit resampling or per-window period alignment. The adapted sequence is then fed into a low-rank segment-level prediction head, which preserves the inductive bias of segmented forecasting while keeping the inter-segment mapping controlled.

The main contributions of this work are summarized as follows.

- We propose DSR-TSF, a period-aware framework for long-horizon time series forecasting that addresses the mismatch between fixed aggregation strategies and evolving periodic states under multi-period superposition and spectral non-stationarity.
- We introduce a spectrum-conditioned routing mechanism that uses dominant frequency amplitudes and normalized frequency positions to characterize the current periodic state, and generates sample- and channel-level weights to adaptively combine multiple period-response branches.
- Controlled ablations show that the gain is not explained by kernel-size matching, static branch priors, or increased head capacity, while routing diagnostics indicate input-dependent branch allocation under changing spectral states.

2 Related work

2.1 Periodic structures and segmented forecasting

Periodicity provides an important inductive bias for long-horizon time series forecasting [Wang et al., 2026b] by organizing temporal dynamics into intra-period and inter-period dependencies. TimesNet [Wu et al., 2023] models multi-periodic temporal variations through two-dimensional representations, while TimeMixer [Wang et al., 2024] captures temporal variations through decomposable multi-scale mixing. LightGTS [Wang et al., 2025] suggests that consistent periodic modeling can support lightweight general forecasting.

These studies demonstrate that periodic unfolding, multi-scale mixing, segmentation, and cross-period mapping are effective for reducing the computational burden of long-horizon forecasting. However, some periodic modeling methods still depend on fixed unfolding, scale mixing, or segmentation schemes. When the dominant periodic state varies across windows, variables, or samples, a static period response may be misaligned with the current input. DSR-TSF retains the segmented prediction structure, but relaxes the fixed-response assumption by introducing input-dependent period-response aggregation before segment-level forecasting.

2.2 Spectrum-aware and frequency-domain modeling

Spectral information provides a global view of time series by revealing dominant frequencies, energy distributions, and periodic variations that are less explicit in the time domain. FITS [Xu et al., 2023] performs lightweight forecasting through interpolation in the complex frequency domain. FreTS [Yi et al., 2023] exploits the global-view and energy-compaction properties of spectral representations to enhance MLP-based temporal modeling. Recent wavelet-based forecasters further exploit multi-resolution structures to capture global and local temporal patterns [Murad et al., 2025, Zhang et al., 2025, Wang et al., 2026a]. TimeKAN [Huang et al., 2025b] addresses intertwined multi-frequency components through frequency decomposition and frequency-specific representation learning.

FreqCycle [Zhang et al., 2026] further shows that multi-scale time-frequency analysis can improve long-horizon forecasting. Despite their effectiveness, these methods mainly use spectral information as a representation space, decomposition target, or filtering mechanism. DSR-TSF takes a different role for spectral information: it uses compact spectral descriptors as conditioning signals that regulate temporal response aggregation, rather than treating the frequency domain as an alternative prediction space.

2.3 Adaptive aggregation and dynamic routing

Fixed forecasting paths are often insufficient for non-stationary time series, motivating input-adaptive forecasting mechanisms. MoLE [Ni et al., 2024] improves the adaptability of linear predictors to periodic changes through a mixture-of-linear-experts design. SOFTS [Han et al., 2024] enhances multivariate information fusion with a STAR aggregate-redistribute mechanism. DTAF [Lu et al., 2026] addresses non-stationarity from both temporal and frequency domains through temporal stabilization and frequency differencing. TimeStacker [Liu et al., 2025b] captures non-stationary patterns via multilevel temporal observations. These methods indicate that input-dependent expert selection [Xu et al., 2026], information aggregation, and non-stationarity adaptation [Xiao et al., 2026a] can improve robustness beyond fully fixed forecasting paths.

DSR-TSF differs from these adaptive methods in the conditioning signal [Yao et al., 2026], routing granularity, and location of adaptation. Rather than routing among complete forecasting experts or adapting representations after a fixed forecasting structure has been formed, it uses dominant spectral descriptors to guide sample-channel-level period-response aggregation before segment-level prediction.

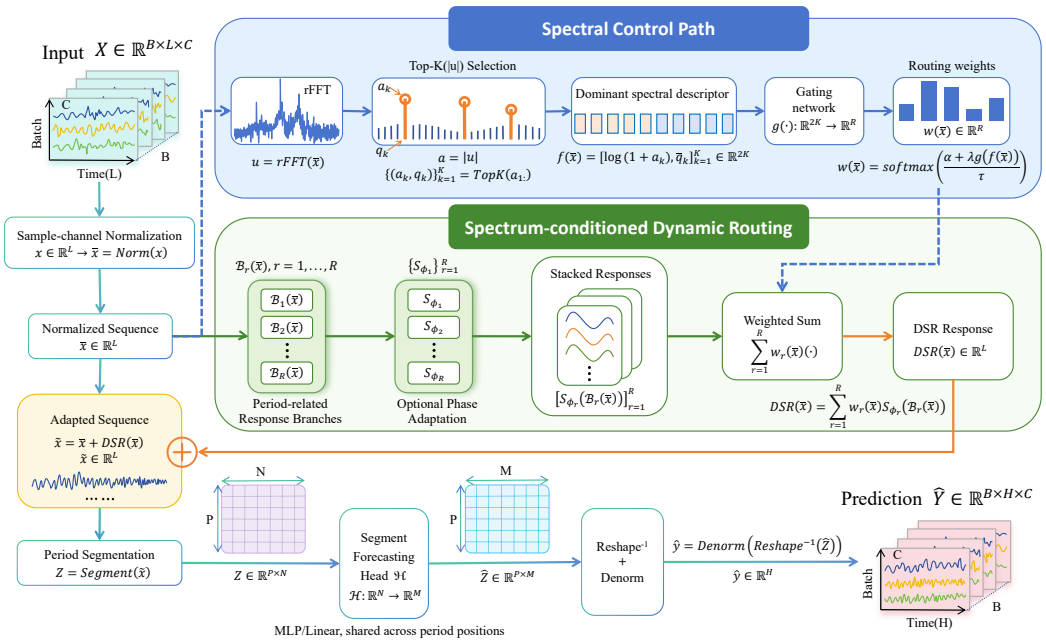


Figure 2: Overall architecture of DSR-TSF.

3 Methodology

3.1 Problem formulation

Long-horizon time series forecasting aims to predict future values over an extended horizon from historical observations [Zhou et al., 2021, Wu et al., 2021]. Let the input and output sequences be denoted as $X \in \mathbb{R}^{B\times L\times C}$ and $Y \in \mathbb{R}^{B\times H\times C}$, where B, L, H, and C denote the batch size, look-back length, prediction horizon, and number of variables, respectively. To enable periodic segmented modeling, we set a fixed period length P and assume that both L and H are divisible by P, yielding $N = L/P$ historical segments and $M = H/P$ future segments. The fixed period length P is used to define the segment-level forecasting grid rather than to assume that all input windows share the same dominant period.

DSR-TSF adapts the representation before segmentation according to the spectral state of each input, while keeping the segmented prediction structure unchanged. Following common channel-independent forecasting models, we present the method for a single sample-channel input $x \in \mathbb{R}^{L}$ and its target $y \in \mathbb{R}^{H}$ by default.

3.2 Overall framework

Based on the above formulation, DSR-TSF introduces input-dependent period-response adaptation before periodic segmented forecasting, allowing the prediction representation to vary with the dominant spectral state of the input.

As shown in Fig. 2, the model first applies sample-channel normalization. The normalized sequence is then processed by a spectral control path, where rFFT and Top-K frequency selection extract dominant spectral descriptors, and a compact gating network maps them into routing weights. These weights are used to adaptively combine multiple period-response branches, optionally with phase adaptation, to produce a DSR response. The response is added back to the normalized sequence, and the resulting adapted sequence is segmented by period and fed into a shared segment-level forecasting head. After inverse reshaping and denormalization, the model outputs the final prediction.

3.3 Spectrum-conditioned dynamic routing

3.3.1 Spectral descriptor construction

DSR-TSF characterizes the current periodic state using dominant spectral descriptors, which are subsequently used to generate dynamic routing weights. Given a single-channel input $x \in \mathbb{R}^L$, we first apply sample-level normalization:

$$\bar{x} = \mathrm{Norm}(x). \tag{1}$$

Here, $\mathrm{Norm}(\cdot)$ denotes mean-centering by default and can be optionally extended to standardization. The normalized sequence is transformed into the frequency domain, and its magnitude spectrum is computed as

$$u = \mathrm{rFFT}(\bar{x}), \quad a = |u|. \tag{2}$$

Since the zero-frequency component mainly reflects the mean level of the input rather than periodic fluctuations, we exclude the DC component during Top-K selection and select the K frequency components with the largest magnitudes from $a_{1:}$:

$$\{(a_k, q_k)\}_{k=1}^{K} = \mathrm{TopK}(a_{1:}), \tag{3}$$

where a_k denotes the magnitude of the k-th selected dominant frequency component, and q_k denotes its frequency-position index. The selected components describe the strongest non-zero spectral responses of the current input, rather than fixed frequency bins shared by all samples. The selected magnitudes and normalized frequency positions are concatenated to construct a compact spectral descriptor:

$$f(\bar{x}) = [\log(1 + a_1), \ldots, \log(1 + a_K), \bar{q}_1, \ldots, \bar{q}_K] \in \mathbb{R}^{2K}, \tag{4}$$

where

$$\bar{q}_k = \frac{q_k}{\lfloor L/2 \rfloor - 1}. \tag{5}$$

The logarithmic transform reduces scale variation among spectral magnitudes, while the normalized frequency positions indicate where the dominant components appear in the spectrum. Therefore, $f(\bar{x})$ encodes both the strength and location of the dominant periodic components. This compact descriptor is not used as a forecasting representation directly; instead, it serves as the conditioning signal that controls dynamic routing among period-response branches. Prop A.1 further discusses a mean-shift invariance property of this descriptor when the DC component is excluded.

3.3.2 Spectrum-conditioned dynamic routing

The dominant spectral descriptor $f(\bar{x})$ is used as the conditioning signal for dynamic routing. Let R denote the number of learnable period-response branches. A gating network

$$g(\cdot) : \mathbb{R}^{2K} \to \mathbb{R}^R \tag{6}$$

maps the spectral descriptor to branch-level routing logits. To provide each branch with a learnable global preference, we introduce a branch prior $\alpha \in \mathbb{R}^R$. The input-dependent routing weights are defined as

$$w(\bar{x}) = \mathrm{softmax}\left(\frac{\alpha + \lambda g\left(f(\bar{x})\right)}{\tau}\right), \quad w(\bar{x}) \in \mathbb{R}^R, \tag{7}$$

where λ denotes the scaling factor for the gating output, and τ denotes the softmax temperature. The r-th element of $w(\bar{x})$, denoted as $w_r(\bar{x})$, softly modulates the contribution of the r-th period-response branch.

This formulation decomposes the routing signal into a global component and an input-dependent component. The learnable prior α captures dataset-level branch preferences, while $g(f(\bar{x}))$ adjusts these preferences according to the spectral state of the current input. In the multivariate setting, the same routing procedure is applied to each sample-channel sequence, producing routing weights that vary across both samples and variables. When the spectral condition is removed, the routing weights depend only on the learnable prior and reduce to static branch preferences, corresponding to the Static-Routing ablation in our experiments.

3.3.3 Periodic response branches and phase adaptation

Each period-response branch $\mathcal{B}_r(\cdot)$ is implemented as a learnable temporal response operator. Given the normalized input $\bar{x}$, the output of the r-th branch is defined as

$$z_r = \mathcal{B}_r(\bar{x}) \in \mathbb{R}^L, \quad r = 1, \ldots, R. \tag{8}$$

These branches do not explicitly estimate exact periods, nor is each branch assigned to a predefined physical period. Instead, they serve as learnable temporal response bases before period-based segmentation, whose contributions are softly modulated by spectrum-conditioned routing weights. This allows DSR-TSF to avoid relying on a single fixed temporal response for all inputs.

When phase adaptation is enabled, each branch output is further processed by a branch-specific phase adaptation operator $S_{\phi_r}(\cdot)$ to adjust potential temporal shifts in the response of the r-th branch. When phase adaptation is disabled, $S_{\phi_r}(\cdot)$ reduces to the identity mapping. The aggregated DSR response is defined as

$$\mathrm{DSR}(\bar{x}) = \sum_{r=1}^{R} w_r(\bar{x}) S_{\phi_r}(z_r) = \sum_{r=1}^{R} w_r(\bar{x}) S_{\phi_r}\left(\mathcal{B}_r(\bar{x})\right). \tag{9}$$

Thus, the final response is a spectrum-conditioned soft aggregation of multiple temporal response bases rather than a fixed response path. Props A.2 and A.3 provide boundedness and conditional stability properties of this routed aggregation.

3.4 Segmented forecasting with adapted periodic representations

After spectrum-conditioned dynamic routing, the model obtains an adapted single-channel representation:

$$\tilde{x} = \bar{x} + \mathrm{DSR}(\bar{x}). \tag{10}$$

This residual form preserves the normalized input as the backbone representation while injecting spectrum-conditioned temporal responses before segmentation. DSR-TSF then follows the periodic segmented forecasting paradigm and reshapes $\tilde{x}$ into a segment-level representation with a fixed period length P:

$$Z = \mathrm{Segment}(\tilde{x}) \in \mathbb{R}^{P \times N}. \tag{11}$$

Here, the first dimension corresponds to intra-period positions, while the second dimension indexes historical period segments. Specifically, each row $Z_{p,:} \in \mathbb{R}^N$ represents the historical segment sequence at the fixed intra-period position p. The fixed period length P defines the segment-level forecasting grid, while the preceding routing module adapts the input representation according to its current spectral state.

The segment-level forecasting head $\mathcal{H}$ learns a mapping from historical segments to future segments at each intra-period position:

$$\hat{Z}_{p,:} = \mathcal{H}(Z_{p,:}), \quad \mathcal{H} : \mathbb{R}^N \to \mathbb{R}^M, \quad \hat{Z} \in \mathbb{R}^{P \times M}. \tag{12}$$

In the current implementation, $\mathcal{H}$ is shared across intra-period positions and can be implemented as a low-rank linear mapping or a residual-gated MLP. Sharing $\mathcal{H}$ keeps the segment-level predictor compact and avoids introducing separate forecasting heads for different intra-period positions.

Finally, the model restores the prediction sequence of length H through inverse reshaping and applies denormalization:

$$\hat{y} = \mathrm{Denorm}\left(\mathrm{Reshape}^{-1}(\hat{Z})\right), \quad \hat{y} \in \mathbb{R}^H. \tag{13}$$

The key design choice is not to increase the capacity of the prediction head, but to adapt the representation before segmented forecasting through spectrum-conditioned dynamic routing. The low-rank head serves as a controlled segment-level mapping component, while the main adaptive capability comes from dynamic period-response aggregation before segmentation. A parameter condition for when the rank-constrained head is more compact than a full segment-level mapping is given in Prop A.4. The overall prediction process can be written as

$$\hat{y} = \mathrm{Denorm}\left(\mathrm{Reshape}^{-1}\left(\mathcal{H}\left(\mathrm{Segment}\left(\bar{x} + \mathrm{DSR}(\bar{x})\right)\right)\right)\right). \tag{14}$$

4 Experiments

We evaluate DSR-TSF from four aspects: forecasting performance, key ablations, model cost, and routing behavior. These evaluations are designed to assess its long-horizon forecasting performance, the contribution of spectrum-conditioned dynamic routing, and the accuracy–cost trade-off introduced by adaptive period-response aggregation.

4.1 Experimental setup

Datasets. We evaluate DSR-TSF on seven widely used long-horizon forecasting benchmarks: ETTh1, ETTh2, ETTm1, ETTm2 [Zhou et al., 2021], Electricity, Traffic, and Weather [Qiu et al., 2024]. These datasets cover diverse sampling frequencies, variable dimensions, and periodic structures, enabling evaluation on low-dimensional electricity transformer series, high-dimensional electricity and traffic series, and multivariate meteorological series. Detailed experiments are provided in Appendix C.

Baselines. We compare DSR-TSF with representative long-horizon forecasting baselines in Table 1. SparseTSF [Lin et al., 2025b] is the most direct baseline due to its periodic segmented forecasting paradigm. We further include recent compact or efficient models, including MixLinear [Ma et al., 2026], Xlinear [Chen et al., 2026], TimeBase [Huang et al., 2025a], and TQNet [Lin et al., 2025a], as well as representative strong baselines, including TimeMixer [Wang et al., 2024], iTransformer [Liu et al., 2024], DLinear [Zeng et al., 2023], and TimesNet [Wu et al., 2023]. Both MLP and Linear versions of DSR-TSF and SparseTSF are reported for direct comparison. Results marked with * are reported results taken from TimeBase [Huang et al., 2025a]; other published baselines are adopted from their original papers.

Forecasting protocol and metrics. The look-back length is fixed to $L = 720$, and the prediction horizon is set to $H \in \{96, 192, 336, 720\}$, following common long-horizon forecasting evaluation practice [Zhang et al., 2024]. We use mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics, where lower values indicate better forecasting performance:

$$\mathrm{MSE} = \frac{1}{N_{\mathrm{eval}}} \sum_{i=1}^{N_{\mathrm{eval}}} (\hat{y}_i - y_i)^2, \quad \mathrm{MAE} = \frac{1}{N_{\mathrm{eval}}} \sum_{i=1}^{N_{\mathrm{eval}}} |\hat{y}_i - y_i|. \tag{15}$$

Here, N_{eval} denotes the total number of forecasted points used for evaluation.

4.2 Main results

Table 1: Average multivariate forecasting results over prediction horizons $\{96, 192, 336, 720\}$ on seven benchmarks. Lower MSE and MAE values indicate better performance. The best results are highlighted in **red bold**, and the second-best results are highlighted in blue underline. Full per-horizon results are reported in Appendix Table 6.

Dataset	Ours				Direct segmented				Compact models								Representative strong baselines							
	DSR-TSF				SparseTSF(2025)				MixLinear (2026)		XLinear (2026)		TimeBase (2025)		TQNet (2025)		TimeMixer (2024)		iTransformer* (2024)		DLinear* (2023)		TimesNet* (2023)	
	MLP		Linear		MLP		Linear																	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.393	0.414	**0.390**	**0.412**	0.432	0.437	0.407	0.419	0.395	-	0.425	0.428	0.396	0.415	0.440	0.452	0.447	0.440	0.454	0.467	0.437	0.448	0.505	0.499
ETTh2	**0.333**	**0.379**	0.334	0.380	0.368	0.410	0.344	0.387	0.339	-	0.359	0.391	0.348	0.399	0.367	0.406	0.365	0.395	0.392	0.422	0.479	0.471	0.433	0.455
ETTm1	**0.348**	0.381	0.352	**0.378**	0.365	0.392	0.362	0.382	0.356	-	0.373	0.390	0.356	0.381	0.362	0.388	0.381	0.396	0.371	0.401	0.359	0.383	0.408	0.425
ETTm2	**0.249**	**0.311**	0.250	0.312	0.264	0.327	0.251	0.313	0.251	-	0.270	0.316	0.252	0.314	0.263	0.323	0.275	0.323	0.279	0.338	0.271	0.337	0.300	0.354
Weather	**0.227**	0.273	**0.227**	0.270	0.231	0.276	0.241	0.280	0.240	-	0.237	**0.268**	0.243	0.284	0.240	0.279	0.240	0.272	0.233	0.273	0.243	0.299	0.254	0.293
Electricity	**0.161**	0.257	0.165	0.259	0.163	**0.254**	0.166	0.259	0.168	-	0.168	0.263	0.167	0.258	0.164	0.261	0.182	0.273	0.166	0.264	0.169	0.272	0.238	0.334
Traffic	0.403	0.274	0.412	0.278	**0.396**	**0.264**	0.412	0.278	0.415	-	0.463	0.295	0.418	0.281	0.412	0.289	0.485	0.298	0.407	0.291	0.419	0.285	0.641	0.346

We follow the standard chronological split and report MSE and MAE for all datasets and horizons. Table 1 reports the average multivariate forecasting results of DSR-TSF and representative baselines on seven long-horizon forecasting datasets. Compared with the direct baseline SparseTSF [Lin et al., 2025b], DSR-TSF reduces the average MSE by approximately 2.37% across the seven datasets. The improvement is more consistent on the ETT benchmarks, with average MSE reductions of approximately 4.24%, 3.41%, 3.73%, and 0.40% on ETTh1, ETTh2, ETTm1, and ETTm2, respectively. These results suggest that spectrum-conditioned periodic responses improve the adaptability of fixed-period segmented forecasting models to varying input spectra.

On higher-dimensional or more complex datasets, including Electricity, Traffic, and Weather, DSR-TSF still achieves MSE improvements over SparseTSF, although the margins are relatively modest. This indicates that DSR-TSF mainly strengthens temporal period-response modeling, while inter-variable dependency modeling and absolute error control in high-dimensional settings remain open for further improvement, as also suggested by recent studies on adaptive variable-dependency modeling [Hu et al., 2025].

4.3 Key ablation studies

Table 2: Key ablation results. Each entry reports the average MSE and MAE over four prediction horizons $H \in \{96, 192, 336, 720\}$. Lower values indicate better performance.Full ablation results are reported in Table 8.

Models	DSR-TSF				Kernel-matched				Static routing				w/o low-rank head			
	MLP		Linear		MLP		Linear		MLP		Linear		MLP		Linear	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.393	0.414	0.390	0.412	0.433	0.436	0.403	0.421	0.402	0.417	0.401	0.418	0.404	0.422	0.403	0.420
ETTh2	0.333	0.379	0.334	0.380	0.380	0.417	0.339	0.384	0.339	0.384	0.340	0.384	0.344	0.387	0.340	0.384
ETTm1	0.348	0.381	0.352	0.378	0.372	0.399	0.362	0.383	0.358	0.387	0.361	0.385	0.362	0.394	0.357	0.380
ETTm2	0.249	0.311	0.250	0.312	0.265	0.322	0.251	0.313	0.253	0.314	0.251	0.313	0.254	0.314	0.252	0.313

To assess the contribution of the key designs in DSR-TSF, we conduct ablation studies around three questions: whether the gain mainly comes from changes in the convolutional receptive field, whether input-dependent spectral conditioning is necessary for routing, and whether the improvement depends on a specific prediction head.

As shown in Table 2, the Kernel-matched variant controls for the effect of the convolutional receptive field, which has been shown to be important for time series modeling in recent convolution-based forecasting architectures [Luo and Wang, 2024]. It keeps the single-branch modeling strategy and prediction head of SparseTSF unchanged, while only matching the convolution kernel size to the period-response branches in DSR-TSF. The full DSR-TSF achieves lower average MSE than this variant across the evaluated ETT datasets, indicating that the improvement cannot be explained solely by kernel-size matching or an enlarged receptive field. Instead, the gain is more likely associated with the input-dependent combination of period responses.

Static routing is used to assess the necessity of spectrum-conditioned routing. This variant removes the input spectral condition and retains only global learnable branch preferences. Compared with static routing, the full DSR-TSF obtains lower average errors in most datasets and prediction-head settings, suggesting that fixed branch preferences are insufficient to adapt to sample-varying spectral structures. This supports the role of input-dependent spectral routing in weighting suitable period-response branches.

The w/o low-rank head variant replaces the low-rank segment-level prediction head with a standard prediction head. Under both MLP and Linear settings, the full DSR-TSF generally achieves lower MSE and MAE. This suggests that the low-rank segment-level mapping is not merely a parameter-compression design. Rather, it imposes structural constraints on inter-segment forecasting relations and reduces redundant degrees of freedom introduced by unconstrained segment-level mappings. Complete MSE and MAE results and relative MSE reductions are provided in Appendix D.

4.4 Computational cost analysis

To examine whether the performance gain mainly comes from increasing model scale, we compare DSR-TSF with SparseTSF and the variant without the low-rank head in terms of accuracy, trainable parameters, MACs, and inference time. As shown in Table 3 and Fig. 3, DSR-TSF achieves lower average MSE under a compact parameter budget. Compared with SparseTSF, DSR-TSF reduces the MSE from 0.407 to 0.390 under the Linear head

Table 3: ETTh1 cost statistics. MSE is averaged over four horizons; time is measured in milliseconds.

Model	Head	MSE	Params	MACs	Time
DSR-TSF	Linear	0.390	0.90K	20.22M	1.92
	MLP	0.393	1.52K	23.40M	2.33
SparseTSF	Linear	0.407	0.45K	6.29M	0.35
	MLP	0.432	5.78K	34.31M	0.46
w/o LR	Linear	0.403	0.82K	22.47M	1.79
	MLP	0.404	1.44K	25.65M	2.26

and from 0.432 to 0.393 under the MLP head, indicating that the gain is not simply explained by increasing the number of trainable parameters.

At the same time, DSR-TSF introduces additional forward-pass computation. DSR-TSF increases inference time relative to SparseTSF, while the difference in MACs depends on the prediction head. Therefore, DSR-TSF should be interpreted as parameter-compact rather than computation-minimal. It trades additional runtime computation for input-dependent period-response adaptation, which is preferable when improved forecasting accuracy under a small trainable-parameter budget is more important than minimum latency.

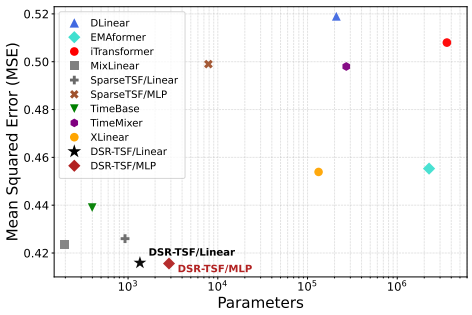


Figure 3: MSE–parameter trade-off on ETTh1 with input length 720 and prediction length 720.

The low-rank segment-level prediction head is used to constrain inter-segment mappings rather than to reduce the whole-model parameter count. Given the segmented representation $Z \in \mathbb{R}^{P\times N}$ and the predicted representation $\hat{Z} \in \mathbb{R}^{P\times M}$, a full-rank segment mapping is

$$\hat{Z} = ZW, \quad W \in \mathbb{R}^{N\times M}, \tag{16}$$

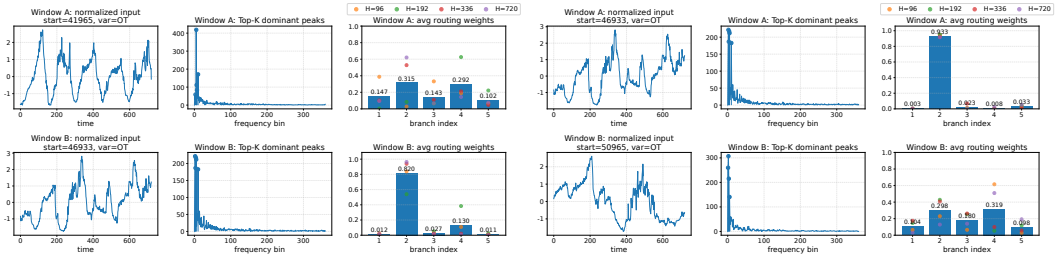
with complexity $\mathcal{O}(NM)$ for each intra-period position. DSR-TSF factorizes the mapping as

$$W \approx AB, \quad A \in \mathbb{R}^{N\times r}, \quad B \in \mathbb{R}^{r\times M}, \tag{17}$$

which reduces the segment-level mapping complexity to $\mathcal{O}(r(N+M))$.

The cost statistics in Table 3 show that DSR-TSF and the w/o low-rank head variant have similar parameter and runtime scales. Therefore, the low-rank head should be interpreted as a structured constraint on segment-level forecasting relations, rather than as a simple parameter-reduction module. Full component-wise complexity is reported in Table 12.

4.5 Mechanism analysis



(a) MLP (b) Linear

Figure 4: Spectral evidence and routing distributions for different Weather windows under MLP and Linear heads. Further routing diagnostics are provided in Figure 8.

To examine whether DSR-TSF performs input-dependent routing rather than collapsing to fixed branch weights, we visualize the spectral evidence chain under different input windows, including the normalized input, the Top-K dominant spectral peaks, and the average routing weights.

As shown in Fig. 4, different Weather windows exhibit distinct temporal patterns and dominant spectral peaks, leading to different routing distributions. Under the Linear-head setting, Window A concentrates most routing mass on the second branch, whereas Window B distributes weights across the second, third, and fourth branches. This suggests that DSR-TSF adjusts its period-response combination according to the current spectral state rather than relying on a fixed branch preference.

The MLP-head setting shows a similar trend: windows with clearer spectral structures tend to activate fewer dominant branches, while windows with more complex local fluctuations rely more on distributed branch collaboration [Xiao et al., 2026b]. Models trained separately for different

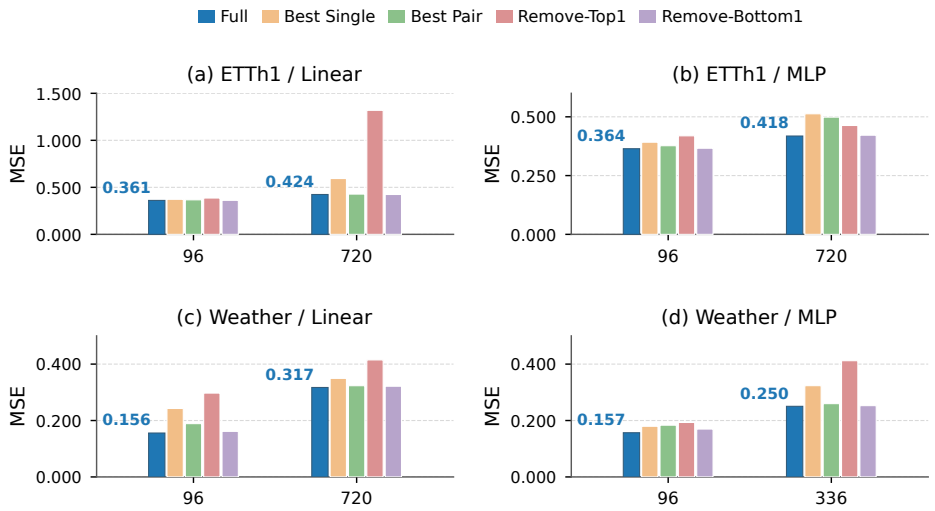


Figure 5: Functional analysis of response branches.

prediction horizons also exhibit distinct routing distributions. Together with the ablation results, these observations provide supporting evidence that spectrum-conditioned routing contributes to adaptive periodic response modeling in DSR-TSF.

We further examine branch functionality through inference interventions on ETTh1 and Weather with Linear and MLP heads (Fig. 5). Best Single and Best Pair retain fixed subsets selected on validation data; Remove-Top1 and Remove-Bottom1 discard the branch with the highest and lowest routing weight, respectively, for each sample and variable. All parameters remain fixed, and retained routing weights are renormalized. Averaged over three seeds, Full achieves lower test MSE than both fixed subsets across all eight configurations, while Remove-Top1 consistently yields higher mean MSE than Remove-Bottom1. These results support branch complementarity and an association between routing weights and predictive contribution within the trained models.

5 Conclusion

We presented DSR-TSF, a spectrum-driven dynamic routing model for long-horizon time series forecasting. DSR-TSF characterizes the current periodic state of the input sequence using dominant frequency magnitudes and normalized frequency positions, and adaptively combines multiple period-response branches through dynamic routing. This design mitigates the mismatch between fixed-period aggregation and evolving spectral structures under multi-period superposition and spectral non-stationarity. By combining periodic segmented forecasting with a low-rank segment-level prediction head, DSR-TSF improves the modeling of dynamic periodic structures while retaining the inductive bias of segmented forecasting. Experiments validate spectrum-conditioned dynamic periodic modeling from three perspectives: forecasting performance, key component analysis, and routing behavior. These results suggest that dynamic spectral conditioning is a useful direction for improving period-aware long-horizon forecasting models.

Acknowledgment

This work was supported in part by the following: the National Natural Science Foundation of China under Grant Nos. U24A20328, U24A20219, 62272281, the Natural Science Foundation of Shandong Province under Grant No. ZR2026MS1059, the Youth Innovation Technology Project of Higher School in Shandong Province under Grant No. 2023KJ212, and the Yantai Natural Science Foundation under Grant No. 2024JCYJ034.

Competing Interests. The authors declare no competing interests.

References

Xinyang Chen, Huidong Jin, Yu Huang, and Zaiwen Feng. Xlinear: A lightweight and accurate mlp-based model for long-term time series forecasting with exogenous inputs. *arXiv preprint arXiv:2601.09237*, 2026.

Yu Chen, Nathalia Céspedes, and Payam Barnaghi. A closer look at transformers for time series forecasting: Understanding why they work and where they struggle. In *Forty-second International Conference on Machine Learning*, 2025.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37: 64145–64175, 2024.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. *arXiv preprint arXiv:2501.13041*, 2025.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025a.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. *arXiv preprint arXiv:2502.06910*, 2025b.

Junhyeok Kang, Yooju Shin, and Jae-Gil Lee. Vardrop: enhancing training efficiency by reducing variate redundancy in periodic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17742–17749, 2025.

Feifei Kou, Jiahao Wang, Lei Shi, Yuhan Yao, Yawen Li, Suguo Zhu, Zhongbao Zhang, and Junping Du. Cfpt: Empowering time series forecasting through cross-frequency interaction and periodic-aware timestamp modeling. In *Forty-second International Conference on Machine Learning*, 2025.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. *arXiv preprint arXiv:2505.12917*, 2025a.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and CL Philip Chen. Sparsetsf: Lightweight and robust time series forecasting via sparse modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. Timebridge: Non-stationarity matters for long-term time series forecasting. In *International Conference on Machine Learning*, pages 39815–39840. PMLR, 2025a.

Qinglong Liu, Cong Xu, Wenhao Jiang, Kaixuan Wang, Lin Ma, and Haifeng Li. Timestacker: A novel framework with multilevel observation for capturing nonstationary patterns in time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025b.

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.

Junkai Lu, Peng Chen, Chenjuan Guo, Yang Shu, Meng Wang, and Bin Yang. Towards non-stationary time series forecasting with temporal stabilization and frequency differencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24070–24078, 2026.

Donghao Luo and Xue Wang. Moderntcn: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.

Aitian Ma, Dongsheng Luo, and Mo Sha. Mixlinear: Extreme low resource multivariate time series forecasting with 0.1k parameters, 2026. URL `https://arxiv.org/abs/2410.02081`.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 19581–19588, 2025.

Ronghao Ni, Zinan Lin, Shuaiqi Wang, and Giulia Fanti. Mixture-of-linear-experts for long-term time series forecasting. In *International conference on artificial intelligence and statistics*, pages 4672–4680. PMLR, 2024.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S Jensen, Zhenli Sheng, et al. Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods. *arXiv preprint arXiv:2403.20150*, 2024.

Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 12640–12648, 2025.

Caixia Wang, Fan Zhang, Xiaofeng Zhang, and Hua Wang. Dtfnet: A dual-modal time-frequency fusion network for non-stationary time series modeling. *Knowledge-Based Systems*, page 116022, 2026a.

Hua Wang, Xianhao Jiao, and Fan Zhang. PESD-TSF: A period-aware and explicit structured decomposition framework for long-term time series forecasting. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 127230–127252. PMLR, 06–11 Jul 2026b. URL `https://proceedings.mlr.press/v306/wang26cs.html`.

Hua Wang, Jinghao Lu, and Fan Zhang. Idealtsf: Can non-ideal data contribute to enhancing the performance of time series forecasting models? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 26224–26232, 2026c.

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. *arXiv preprint arXiv:2405.14616*, 2024.

Yihang Wang, Yuying Qiu, Peng Chen, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. Lightgts: A lightweight general time series forecasting model. *arXiv preprint arXiv:2506.06005*, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis, 2023. *arXiv preprint arXiv:2210.02186*, 2023.

Canran Xiao and Liwei Hou. Prototype-aligned federated soft-prompts for continual web personalization. In *Proceedings of the ACM Web Conference 2026*, pages 6743–6754, 2026.

Canran Xiao and Yongmei Liu. A multifrequency data fusion deep learning model for carbon price prediction. *Journal of Forecasting*, 44(2):436–458, 2025.

Canran Xiao, Tianxiang Xu, Siyuan Ma, Yiyang Jiang, Haoyu Gao, and Yuhan Wu. Reversible primitive–composition alignment for continual vision–language learning. In *International Conference on Learning Representations*, volume 2026, pages 88700–88722, 2026a.

Canran Xiao, Chuangxin Zhao, Zong Ke, Fei Shen, and Li Shen. Curiosity-driven cooperation for long-tailed multi-label learning. *Neural Networks*, page 109075, 2026b.

Mengzhu Xu, Hanzhi Liu, Ningkang Peng, Qianyu Chen, and Canran Xiao. Affordance-first decomposition for continual learning in video-language understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3908–3919, June 2026.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. *arXiv preprint arXiv:2307.03756*, 2023.

Zhen Yao, Bo Lang, Zhihao Zheng, Haoyan Xu, Zhengtao Yao, Zhikang Dong, and Mooi Choo Chuah. From semantic to reasoing: A survey of language-guided segmentation in the multimodal era. 2026.

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. *Advances in Neural Information Processing Systems*, 36:76656–76679, 2023.

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, pages 11121–11128, 2023.

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, pages 28159–28167, 2026.

Jiawen Zhang, Xumeng Wen, Zhenwei Zhang, Shun Zheng, Jia Li, and Jiang Bian. Probts: Benchmarking point and distributional forecasting across diverse prediction horizons. *Advances in Neural Information Processing Systems*, 37:48045–48082, 2024.

Zichi Zhang, Tuan Dung Pham, Yimeng An, Ngoc Phu Doan, Majed Alsharari, Viet-Hung Tran, Anh-Tuan Hoang, Hans Vandierendonck, and Son T Mai. Waveletmixer: a multi-resolution wavelets based mlp-mixer for multivariate long-term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22741–22749, 2025.

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*, volume 35, pages 11106–11115, 2021.

A Theoretical properties of DSR-TSF

This section provides formal properties of the proposed dynamic spectral routing mechanism and the rank-constrained prediction head. These results are intended to clarify structural behaviors of DSR-TSF, rather than to establish generalization guarantees or universal forecasting error bounds. The notation follows the methodology section: $x \in \mathbb{R}^L$ denotes a raw single-channel input window, $\bar{x} = \mathrm{Norm}(x)$ denotes the normalized sequence used for spectral descriptor construction, K denotes the number of retained dominant frequency components, and R denotes the number of period-response branches.

A.1 Mean-shift invariance of the spectral descriptor

Following Eq. (2) and Eq. (3), DSR-TSF computes the amplitude spectrum from the normalized input $\bar{x}$, excludes the DC component, and selects the Top-K dominant non-zero frequency components $\{(a_k, q_k)\}_{k=1}^{K}$ from $a_{1:}$. Here, a_k denotes the magnitude of the k-th selected dominant frequency component, and q_k denotes its frequency-position index.

Proposition A.1 (Mean-shift invariance of the spectral descriptor)**.** *Let $\bar{x} \in \mathbb{R}^L$ be the normalized input sequence used for spectral descriptor construction, and let $\bar{x}' = \bar{x} + c\mathbf{1}$, where $c \in \mathbb{R}$ and $\mathbf{1} \in \mathbb{R}^L$ is the all-one vector. Let*

$$u = \mathrm{rFFT}(\bar{x}), \qquad u' = \mathrm{rFFT}(\bar{x}').$$

If the spectral descriptor is constructed from the non-zero frequency magnitudes by excluding the DC component, then the selected Top-K non-zero frequency indices and their magnitudes are invariant to the additive shift $c\mathbf{1}$.

Proof. By the linearity of the real Fourier transform, we have

$$u' = \mathrm{rFFT}(\bar{x} + c\mathbf{1}) = \mathrm{rFFT}(\bar{x}) + c\,\mathrm{rFFT}(\mathbf{1}).$$

The real Fourier transform of a constant sequence is non-zero only at the zero-frequency component. Therefore, for every non-zero frequency index $q > 0$,

$$u'_q = u_q.$$

Taking magnitudes gives

$$|u'_q| = |u_q|, \qquad q = 1, \ldots, F-1.$$

Since the Top-K operation in Eq. (3) is applied only to the non-zero magnitude spectrum, the selected frequency indices and their corresponding magnitudes remain unchanged after adding the constant shift $c\mathbf{1}$. This proves the claim. □

This proposition explains why removing the DC component allows the spectral descriptor to focus on periodic fluctuations rather than the absolute mean level of the normalized input window.

A.2 Boundedness of dynamic period-response routing

Following Eq. (8) and Eq. (9), DSR-TSF aggregates R period-response branches through spectrum-conditioned routing weights. For notational compactness, we define the effective response of the r-th branch as

$$\mathcal{G}_r(\bar{x}) = S_{\phi_r}\left(\mathcal{B}_r(\bar{x})\right),$$

where $\mathcal{B}_r(\bar{x})$ is the raw response of the r-th branch and S_{ϕ_r} is the optional phase-adaptation operator. When phase adaptation is disabled, S_{ϕ_r} reduces to the identity mapping and $\mathcal{G}_r(\bar{x}) = \mathcal{B}_r(\bar{x})$. Therefore, Eq. (9) can be equivalently written as

$$\mathrm{DSR}(\bar{x}) = \sum_{r=1}^{R} w_r(\bar{x})\mathcal{G}_r(\bar{x}).$$

The routing weights are produced by the softmax routing rule in Eq. (7), and thus satisfy

$$w_r(\bar{x}) \geq 0, \qquad \sum_{r=1}^{R} w_r(\bar{x}) = 1.$$

Proposition A.2 (Boundedness of dynamic period-response routing)**.** *Assume that the routing weights satisfy* $w_r(\bar{x}) \geq 0$ *and* $\sum_{r=1}^{R} w_r(\bar{x}) = 1$. *Then the DSR response satisfies*

$$\|\mathrm{DSR}(\bar{x})\| \leq \sum_{r=1}^{R} w_r(\bar{x})\|\mathcal{G}_r(\bar{x})\| \leq \max_{1\leq r\leq R} \|\mathcal{G}_r(\bar{x})\|.$$

Proof. Using the triangle inequality and the non-negativity of the routing weights, we obtain

$$\|\mathrm{DSR}(\bar{x})\| = \left\| \sum_{r=1}^{R} w_r(\bar{x})\mathcal{G}_r(\bar{x}) \right\| \leq \sum_{r=1}^{R} w_r(\bar{x})\|\mathcal{G}_r(\bar{x})\|.$$

Since the routing weights sum to one, the weighted average of the branch-response norms is upper bounded by the maximum branch-response norm:

$$\sum_{r=1}^{R} w_r(\bar{x})\|\mathcal{G}_r(\bar{x})\| \leq \max_{1\leq r\leq R} \|\mathcal{G}_r(\bar{x})\|.$$

This completes the proof. □

Prop A.2 shows that the DSR response is a convex combination of effective period-response branches. Thus, the routing mechanism adaptively combines branch responses without producing an output whose norm exceeds the largest effective branch response for the same normalized input.

A.3 Stability of routed aggregation under input perturbations

We further provide a conditional stability result for the routed aggregation. This result does not claim that DSR-TSF always improves forecasting accuracy. Instead, it states that if the effective branch responses and the routing map are Lipschitz continuous, then the routed aggregation is also Lipschitz continuous.

Proposition A.3 (Lipschitz stability of routed aggregation)**.** *Assume that each effective branch response $\mathcal{G}_r$ is L_G-Lipschitz and uniformly bounded by M_G, i.e.,*

$$\|\mathcal{G}_r(\bar{x}) - \mathcal{G}_r(\bar{y})\| \le L_G\|\bar{x} - \bar{y}\|, \qquad \|\mathcal{G}_r(\bar{x})\| \le M_G,$$

for all $r = 1, \dots, R$. Assume further that the routing map $w(\bar{x}) = [w_1(\bar{x}), \dots, w_R(\bar{x})]$ is L_w-Lipschitz in the ℓ_1 norm:

$$\|w(\bar{x}) - w(\bar{y})\|_1 \le L_w\|\bar{x} - \bar{y}\|.$$

Then the DSR response is Lipschitz continuous and satisfies

$$\|\mathrm{DSR}(\bar{x}) - \mathrm{DSR}(\bar{y})\| \le (L_G + M_G L_w)\,\|\bar{x} - \bar{y}\|.$$

Proof. Using the equivalent DSR response form derived from Eq. (9), we decompose the difference between two DSR responses as

$$\mathrm{DSR}(\bar{x}) - \mathrm{DSR}(\bar{y}) = \sum_{r=1}^{R} w_r(\bar{x})\,(\mathcal{G}_r(\bar{x}) - \mathcal{G}_r(\bar{y})) + \sum_{r=1}^{R} (w_r(\bar{x}) - w_r(\bar{y}))\,\mathcal{G}_r(\bar{y}).$$

For the first term, using the convexity of the routing weights and the L_G-Lipschitz continuity of each effective branch response,

$$\left\| \sum_{r=1}^{R} w_r(\bar{x})\,(\mathcal{G}_r(\bar{x}) - \mathcal{G}_r(\bar{y})) \right\| \le \sum_{r=1}^{R} w_r(\bar{x}) L_G\|\bar{x} - \bar{y}\| = L_G\|\bar{x} - \bar{y}\|.$$

For the second term, using the uniform branch-response bound and the ℓ_1-Lipschitz condition of the routing map,

$$\left\| \sum_{r=1}^{R} (w_r(\bar{x}) - w_r(\bar{y}))\,\mathcal{G}_r(\bar{y}) \right\| \le \sum_{r=1}^{R} |w_r(\bar{x}) - w_r(\bar{y})|\|\mathcal{G}_r(\bar{y})\| \le M_G\|w(\bar{x}) - w(\bar{y})\|_1 \le M_G L_w\|\bar{x} - \bar{y}\|.$$

Combining the two bounds gives

$$\|\mathrm{DSR}(\bar{x}) - \mathrm{DSR}(\bar{y})\| \le (L_G + M_G L_w)\,\|\bar{x} - \bar{y}\|.$$

This proves the proposition. □

This result indicates that the dynamic routing operation remains stable under small perturbations of the normalized input when the effective branch responses and the routing network are well behaved.

A.4 Parameter condition for the rank-constrained prediction head

Following Eqs. (10)–(12), DSR-TSF first obtains the adapted representation $\tilde{x}$, reshapes it into $Z \in \mathbb{R}^{P \times N}$, and applies a shared segment-level forecasting head $\mathcal{H} : \mathbb{R}^N \to \mathbb{R}^M$ to predict $\hat{Z} \in \mathbb{R}^{P \times M}$. The full-rank segment-level mapping is given in Eq. (16), and its rank-constrained factorization is given in Eq. (17).

Proposition A.4 (Parameter condition for the rank-constrained prediction head)**.** *A full segment-level linear mapping $W \in \mathbb{R}^{N \times M}$ requires NM weight parameters. A rank-constrained factorized head with $A \in \mathbb{R}^{N \times r}$ and $B \in \mathbb{R}^{r \times M}$ requires $r(N + M)$ weight parameters. The factorized head uses fewer parameters than the full head if and only if*

$$r < \frac{NM}{N + M}.$$

Proof. The full segment-level mapping in Eq. (16) contains NM weight parameters. The factorized mapping in Eq. (17) contains

$$Nr + rM = r(N + M),$$

weight parameters. Therefore, the factorized head uses fewer parameters precisely when

$$r(N + M) < NM.$$

Since $N + M > 0$, dividing both sides by $N + M$ gives

$$r < \frac{NM}{N + M}.$$

This proves the claim. □

Prop A.4 clarifies when the rank-constrained head leads to parameter savings. In particular, the compression effect depends on the ratio between the rank r and the segment dimensions $N = L/P$ and $M = H/P$. Therefore, the rank-constrained head is most parameter-efficient when the segment-level input and output dimensions are sufficiently larger than r, such as under smaller period lengths or longer forecasting horizons.

A.5 Discussion

The above propositions establish structural properties of DSR-TSF under explicit assumptions. Prop A.1 shows that removing the DC component makes the spectral descriptor insensitive to additive mean shifts of the normalized input. Prop A.2 shows that the DSR response is bounded by the effective branch-response envelope because the routing weights form a simplex. Prop A.3 provides a conditional perturbation-stability result for the dynamic routing aggregation. Prop A.4 gives an explicit condition under which the rank-constrained prediction head reduces the number of parameters relative to a full dense segment-level prediction head.

These results should be interpreted as formal properties of the proposed architecture, not as guarantees of universally lower forecasting error. The empirical results and ablation studies in the main paper evaluate whether these structural properties translate into improved forecasting performance on standard benchmarks.

B Failure modes and boundary analysis

This appendix discusses the boundary conditions and potential failure modes of DSR-TSF. The goal is not to claim that spectrum-conditioned routing is universally beneficial, but to clarify when the proposed mechanism is expected to help, when its benefit may diminish, and how such cases can be diagnosed in practice.

B.1 Scope of spectrum-conditioned periodic adaptation

DSR-TSF is designed to improve lightweight segmented forecasting under dynamic periodic structures. Its core mechanism extracts compact spectral descriptors from the input window, as defined in Eqs. (2)–(4), and uses them to generate routing weights over learnable period-response branches through Eq. (7). This design is most suitable when the forecasting error is partly caused by window-level spectral variation, mixed periodicity, or mismatch between a fixed aggregation path and the current periodic state.

However, this scope also defines the boundary of the method. DSR-TSF mainly addresses temporal periodic-response mismatch, rather than all forms of non-stationarity. When the forecasting error is dominated by abrupt events, structural breaks, external interventions, or long-term distributional drift, the spectral state observed in the historical window may provide insufficient evidence for future changes. In such cases, dynamic routing can still adjust the period-response aggregation, but it may not fully compensate for non-periodic or externally driven variations.

Therefore, DSR-TSF should be interpreted as a compact mechanism for improving adaptation to dynamic spectral structures within a segmented forecasting framework, rather than as a universal solution for non-stationary time series forecasting.

B.2 Main failure modes

Persistent spectral mismatch. A primary failure mode occurs when the dominant spectral state extracted from the input window is no longer representative of the forecasting horizon. The routing module assumes that the current spectral descriptor provides useful evidence for selecting period-response branches. Under abrupt regime changes, intervention-driven shifts, or persistent spectral drift, the future sequence may follow a different periodic pattern from the observed window. In this case, the model can still produce input-dependent routing weights, but the spectral evidence used for routing may become mismatched with the target horizon.

Weak or non-periodic temporal structure. DSR-TSF is most effective when periodic components contribute meaningfully to the forecasting target. If the sequence is dominated by random fluctuations, sparse events, monotonic trends, or weak periodicity, the extracted Top-K spectral descriptor may provide limited routing evidence. The routing module may then behave similarly to a static branch preference, and the gain from spectrum-conditioned routing can become marginal. This does not invalidate the design, but indicates that its advantage depends on whether periodic structure explains a meaningful part of the forecasting error.

Strong cross-variable dependency. The current DSR-TSF design mainly focuses on channel-wise temporal periodic adaptation. This keeps the model lightweight, but limits its ability to explicitly capture complex cross-variable dependencies. In high-dimensional datasets, future values may depend on spatial propagation, delayed interactions, or coordinated changes across variables. When such dependencies dominate the forecasting error, channel-wise periodic adaptation alone may be insufficient. This boundary is consistent with the design goal of DSR-TSF, which primarily targets temporal periodic mismatch rather than full cross-variable relation modeling.

Dependence on a predefined period length. DSR-TSF relies on a predefined period length P to reshape the adapted sequence into segment-level representations, as defined in Eq. (11). This design reduces the complexity of long-horizon forecasting, but also introduces a period prior. If P is well aligned with the dominant temporal scale, the segment-level head can learn more stable inter-segment mappings. If P is poorly chosen, the resulting segment alignment may weaken both dynamic response aggregation and segment-level forecasting. Although spectrum-conditioned routing improves adaptation within the fixed segmentation structure, it does not fully remove the dependence on a reasonable period prior.

B.3 Controlled evaluation under periodic regime shifts

Table 4: Controlled evaluation of periodic dominance transitions. Results report test MSE as mean $\pm$ sample standard deviation over five random seeds.

Scenario	H	Segmented	DSR-TSF routing interventions			
			Uniform	Mean	Shuffled	Native
Stable	96	0.017 ± 0.001	0.102 ± 0.076	0.009 ± 0.001	0.009 ± 0.001	$\mathbf{0.009} \pm 0.001$
	192	0.019 ± 0.003	0.197 ± 0.108	0.020 ± 0.004	$\mathbf{0.018} \pm 0.003$	0.019 ± 0.003
Gradual	96	0.173 ± 0.002	0.182 ± 0.064	0.098 ± 0.006	0.100 ± 0.012	$\mathbf{0.093} \pm 0.008$
	192	0.178 ± 0.004	0.269 ± 0.075	0.130 ± 0.020	0.136 ± 0.034	$\mathbf{0.120} \pm 0.016$
Observed	96	0.027 ± 0.002	0.101 ± 0.074	0.022 ± 0.004	0.021 ± 0.002	$\mathbf{0.021} \pm 0.002$
	192	$\mathbf{0.031} \pm 0.002$	0.202 ± 0.106	0.043 ± 0.011	0.041 ± 0.008	0.041 ± 0.008
Future-only	96	0.842 ± 0.005	$\mathbf{0.786} \pm 0.064$	0.892 ± 0.010	0.911 ± 0.032	0.896 ± 0.017
	192	0.831 ± 0.009	$\mathbf{0.800} \pm 0.107$	0.868 ± 0.020	0.893 ± 0.018	0.890 ± 0.017

We examine whether spectral routing remains useful when periodic dominance changes. Synthetic univariate sequences combine two periodic components drawn from $\{12, 20, 36, 60\}$, with $L = 720$, $P = 24$, $H \in \{96, 192\}$, and phase adaptation disabled. Stable preserves their relative dominance; Gradual progressively exchanges dominance within history; Observed introduces an abrupt switch before the forecasting boundary; and Future-only changes dominance only in the prediction interval. Training and validation contain 8,000 and 1,600 sequences from the first three scenarios, respectively; each test scenario contains 2,400 sequences.

For each seed and horizon, four routing interventions share the same trained DSR-TSF checkpoint and fixed branch and prediction-head parameters. Native retains the original route; Mean uses the training-set average route; Shuffled exchanges routes across samples with different historical dominant periods; and Uniform assigns equal branch weights. The separately trained Segmented baseline uses the same prediction head with response aggregation removed.

Table 4 shows the clearest mean advantage of Native routing under Gradual transitions. The improvement narrows under Observed transitions, where the Segmented baseline performs better at $H = 192$. Uniform achieves the lowest mean MSE in both Future-only settings. These results support spectral routing when informative periodic evolution is visible in history, while identifying abrupt and historically unobserved transitions as important boundaries of its predictive benefit.

B.4 Practical diagnostics

The above boundary analysis suggests several practical diagnostics. First, spectral reliability can be examined by inspecting the stability and sharpness of dominant frequency peaks across input windows. Flat spectra or highly unstable peaks indicate that the spectral descriptor may provide weak routing evidence. Second, routing-deviation patterns can be used to verify whether the model adjusts branch weights across samples and variables. Nearly identical routing weights may indicate a collapse toward static branch preferences, whereas highly unstructured variations may suggest noisy conditioning signals.

Third, sensitivity to K and τ should be reported when applying DSR-TSF to new datasets. A broad low-error region indicates robust routing behavior, while a sharp optimum suggests stronger dependence on careful tuning. Finally, the period length P should be checked against the sampling frequency and domain prior. Strong sensitivity to P indicates that adaptive period selection may be needed.

B.5 Future directions and scope of the analysis

The identified failure modes suggest several future directions. Adaptive period selection may reduce the dependence on a predefined P, either by selecting the segment length from spectral evidence or by combining multiple segment scales. Online spectral adaptation with explicit drift detection may help under persistent spectral mismatch, although it should be designed carefully to avoid test leakage and overreaction to short-term noise. Lightweight cross-variable interaction, such as sparse variable selection, low-rank variable mixing, or local variable grouping, may further improve performance on high-dimensional datasets where inter-variable dependency dominates the forecasting error.

This appendix provides a boundary analysis rather than a full theoretical characterization of DSR-TSF. It isolates the main assumptions behind spectrum-conditioned routing and segmented forecasting, and clarifies when these assumptions may weaken. The analysis does not claim that DSR-TSF is universally optimal for all non-stationary time series. Instead, it identifies the regimes where the proposed mechanism is expected to help and the conditions under which additional components may be required.

B.6 Broader impacts.

DSR-TSF may benefit long-horizon forecasting applications such as energy management, transportation planning, weather analysis, and industrial monitoring by improving the accuracy–efficiency trade-off. However, inaccurate forecasts may affect downstream decisions in resource- or safety-critical settings. Since the model relies on historical observations and does not explicitly handle rare events, external interventions, or distribution shifts, it should be deployed with dataset-specific validation, error monitoring, and human oversight in high-stakes applications.

C Experimental details

C.1 Experimental protocol

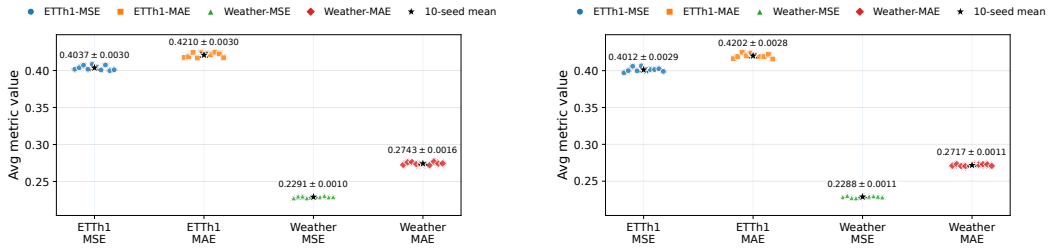
All datasets are split chronologically into training, validation, and test sets. No random split is applied to the raw time series, since it would leak future observations into training. Normalization statistics

are computed only from the training split and then applied to the validation and test splits. Samples are constructed by sliding windows with look-back length L and prediction horizon H.

Models are trained independently for each prediction horizon. We use AdamW to minimize MSE and select the checkpoint with the lowest validation loss for final test evaluation. Early stopping is applied based on validation performance, and the test set is reserved only for final reporting. Unless otherwise specified, DSR-TSF, SparseTSF, and all controlled variants use the same data splits, input length, prediction horizons, training objective, checkpoint selection criterion, and evaluation metrics. Experiments on the ETT-series and Weather datasets are conducted on an NVIDIA GeForce RTX 3050 GPU, whereas Electricity and Traffic experiments are conducted on an NVIDIA A800 80GB GPU to accommodate their higher dimensionality. Hardware differences are used only to ensure feasible training and do not affect the data splits, model selection criterion, or evaluation metrics.

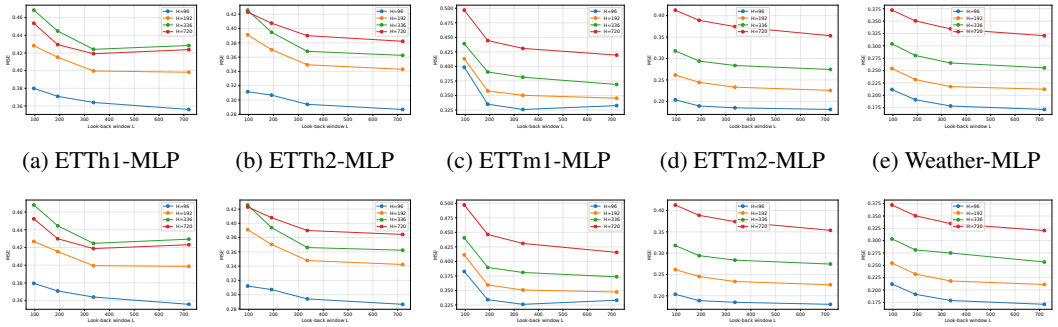
We report MSE and MAE on the original scale after inverse normalization, using the metrics defined in Eq. (15). The training loss is

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{BHC} \sum_{b=1}^{B} \sum_{t=1}^{H} \sum_{c=1}^{C} \left(\hat{Y}_{b,t,c} - Y_{b,t,c} \right)^2 . \tag{18}$$



(a) MLP (b) Linear

Figure 6: Random seed robustness of DSR-TSF on ETTh1 and Weather. Each colored marker denotes the average result over four prediction horizons under one random seed. Black stars denote the 10-seed mean, and the annotations report the corresponding mean ± standard deviation.



(a) ETTh1-MLP (b) ETTh2-MLP (c) ETTm1-MLP (d) ETTm2-MLP (e) Weather-MLP

(f) ETTh1-Linear (g) ETTh2-Linear (h) ETTm1-Linear (i) ETTm2-Linear (j) Weather-Linear

Figure 7: Look-back window sensitivity of DSR-TSF across datasets and prediction heads. Each subfigure reports MSE under different look-back lengths $L \in \{96, 192, 336, 720\}$ across four prediction horizons $H \in \{96, 192, 336, 720\}$. The first row shows the MLP-head results, and the second row shows the Linear-head results.

Random seed robustness. To examine whether the reported performance is sensitive to random initialization, we further evaluate DSR-TSF with ten random seeds $\{2023, \ldots, 2032\}$ on ETTh1 and Weather. For each seed, we compute the average MSE and MAE over four prediction horizons $H \in \{96, 192, 336, 720\}$, and visualize the seed-wise results for both the MLP and Linear prediction heads in Fig. 6. The colored markers denote individual seed runs, while the black stars and annotations report the corresponding mean ± standard deviation. The small dispersion around the mean suggests that DSR-TSF is not strongly dependent on a particular random initialization.

Look-back window selection. We also examine the effect of the look-back window length L by evaluating $L \in \{96, 192, 336, 720\}$ on ETT and Weather benchmarks. As shown in Fig. 7, increasing L generally provides more historical context and often leads to competitive or improved performance, although the sensitivity varies across datasets, horizons, and prediction heads. Based on this analysis, we use $L = 720$ as the default look-back length in the main experiments, which provides a consistent and sufficiently long historical context while avoiding dataset-specific look-back tuning.

C.2 Hyperparameter sensitivity analysis

Table 5: Hyperparameter sensitivity analysis on ETTh1. We report MSE under four prediction horizons. For each hyperparameter group, Mean denotes the average MSE across the tested values, and Dev. denotes the maximum absolute deviation from the mean.

Horizon	Sensitivity to K							Sensitivity to τ						
	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	Mean	Dev.	$\tau=0.5$	$\tau=0.75$	$\tau=1.0$	$\tau=1.5$	$\tau=2.0$	Mean	Dev.
96	0.357	0.358	**0.350**	0.353	0.353	0.3542	0.004	0.3550	**0.3518**	0.3525	0.3557	0.3558	0.3530	0.002
192	0.397	0.399	**0.391**	0.397	0.395	0.3958	0.004	0.399	**0.393**	0.399	0.399	0.397	0.3960	0.003
336	0.419	0.417	**0.416**	0.422	0.417	0.4190	0.003	**0.416**	**0.416**	**0.416**	0.417	0.417	0.4160	0.001
720	0.419	0.417	**0.415**	0.422	0.419	0.4184	0.004	0.4175	0.4184	**0.415**	0.4184	0.4181	0.4165	0.002

Table 5 shows that DSR-TSF is stable with respect to both K and τ within the tested ranges. When K varies from 3 to 7, the maximum deviation from the mean MSE is no larger than 0.004 across all horizons. This suggests that the model does not rely on a single carefully tuned value of K. Among the tested values, $K = 5$ achieves the lowest MSE across the four horizons, indicating that a moderate number of dominant frequency components is sufficient to provide effective routing evidence. Increasing K further does not lead to consistent gains, suggesting that additional low-energy components may introduce weak or noisy spectral evidence.

The sensitivity to τ is also mild. Across the tested temperatures, the maximum deviation from the mean MSE ranges from 0.001 to 0.003. A smaller τ encourages sharper branch specialization, whereas a larger τ produces smoother multi-branch aggregation. The results show that moderate temperatures generally maintain stable performance, with $\tau = 0.75$ performing slightly better for shorter horizons and $\tau = 1.0$ achieving the lowest error for the longest horizon.

Overall, these results indicate that the gains of DSR-TSF are not an artifact of fragile hyperparameter tuning. Together with the ablation studies, the sensitivity analysis supports the robustness of spectrum-conditioned routing under reasonable choices of its key hyperparameters.

C.3 Full forecasting results

Table 6 reports the full multivariate forecasting results across all datasets and prediction horizons. To ensure a controlled comparison for the proposed mechanism, DSR-TSF, SparseTSF, and all ablation variants are trained and evaluated under the same implementation, data protocol, input length, prediction horizons, optimization objective, checkpoint selection criterion, and evaluation metrics. We therefore use these reproduced results as the primary evidence for evaluating the effect of spectrum-conditioned dynamic routing.

The table also includes representative published baselines for broader context. TimeBase and the entries marked with * are taken from TimeBase [Huang et al., 2025a], while the remaining published results are adopted from the corresponding original papers or public reports when available. Because these results may differ in look-back length, implementation details, training schedules, and hyperparameter selection protocols, they are used only as reference points for positioning DSR-TSF. The main empirical claims are based on the reproduced SparseTSF comparison and the controlled ablation studies under a shared protocol.

C.4 Additional evaluation on high-dimensional datasets

We further evaluate the channel-independent DSR-TSF backbone on METR-LA, PM2.5, and Temp to examine its performance across additional multivariate forecasting tasks. These datasets contain 184–207 variables. Both Linear and MLP prediction heads are evaluated for $H \in \{96, 192, 336, 720\}$.

Table 6: Full multivariate forecasting results under input length $L = 720$ and prediction horizons $H \in \{96, 192, 336, 720\}$. Lower MSE and MAE values indicate better performance. The top three results in each row and each metric are highlighted in black bold.

Dataset	Horizon	Ours				Direct segmented				Compact models								Representative strong baselines							
		DSR-TSF				SparseTSF(2025)				MixLinear (2026)		Xlinear (2026)		TimeBase (2025)		TQNet (2025)		TimeMixer (2024)		iTransformer* (2024)		DLinear* (2023)		TimesNet* (2023)	
		MLP		Linear		MLP		Linear																	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.35**	**0.384**	**0.347**	**0.381**	0.372	0.395	0.362	0.389	0.351	-	0.369	0.393	**0.349**	**0.384**	0.379	0.404	0.375	0.400	0.389	0.421	0.378	0.402	0.437	0.454
	192	**0.391**	**0.407**	**0.387**	**0.404**	0.412	0.417	0.404	0.412	0.395	-	0.421	0.423	**0.387**	**0.410**	0.429	0.441	0.429	0.421	0.424	0.446	0.415	0.425	0.456	0.469
	336	0.416	**0.420**	**0.409**	**0.423**	0.444	0.439	0.435	0.428	**0.411**	-	0.455	0.440	**0.408**	**0.418**	0.454	0.455	0.484	0.458	0.456	0.469	0.449	0.449	0.494	0.494
	720	**0.415**	**0.445**	**0.415**	**0.441**	0.499	0.497	0.426	0.448	**0.423**	-	0.453	0.456	0.439	**0.446**	0.499	0.506	0.498	0.482	0.545	0.532	0.507	0.517	0.632	0.578
	Avg.	**0.393**	**0.414**	**0.390**	**0.412**	0.432	0.437	0.407	0.419	**0.395**	-	0.425	0.428	0.396	**0.415**	0.440	0.452	0.447	0.440	0.454	0.467	0.437	0.448	0.505	0.499
ETTh2	96	**0.271**	**0.333**	**0.273**	**0.334**	0.304	0.359	0.294	0.346	**0.283**	-	0.286	**0.337**	0.292	0.350	0.288	0.354	0.289	0.341	0.305	0.361	0.294	0.360	0.349	0.403
	192	**0.331**	**0.370**	**0.331**	**0.372**	0.359	0.397	0.340	**0.377**	**0.336**	-	0.363	0.388	0.341	0.387	0.377	0.403	0.372	0.392	0.405	0.421	0.412	0.437	0.500	0.488
	336	**0.351**	**0.392**	**0.355**	**0.394**	0.387	0.424	0.360	**0.398**	**0.355**	-	0.378	0.407	0.358	0.410	0.377	0.415	0.386	0.414	0.411	0.436	0.471	0.478	0.445	0.465
	720	**0.377**	**0.421**	**0.376**	**0.420**	0.423	0.460	0.383	**0.425**	**0.380**	-	0.408	0.431	0.400	0.448	0.424	0.452	0.412	0.434	0.448	0.470	0.740	0.609	0.438	0.465
	Avg.	**0.333**	**0.379**	**0.334**	**0.380**	0.368	0.410	0.344	**0.387**	**0.339**	-	0.359	0.391	0.348	0.399	0.367	0.406	0.365	0.395	0.392	0.422	0.479	0.471	0.433	0.455
ETTm1	96	**0.292**	0.352	0.309	**0.351**	0.309	0.362	0.313	0.354	0.308	-	0.311	**0.351**	0.310	0.354	**0.296**	**0.349**	0.320	0.357	0.315	0.369	**0.307**	**0.350**	0.359	0.391
	192	**0.333**	**0.372**	**0.332**	**0.366**	0.346	0.381	0.347	0.376	**0.337**	-	0.353	0.376	0.338	**0.371**	**0.337**	0.374	0.361	0.381	0.349	0.388	0.347	0.381	0.368	0.398
	336	**0.360**	0.390	**0.360**	**0.383**	0.373	0.398	0.367	**0.386**	0.365	-	0.382	0.398	**0.364**	**0.386**	0.369	0.393	0.390	0.404	0.381	0.409	0.367	0.387	0.429	0.438
	720	**0.407**	**0.411**	**0.406**	**0.410**	0.430	0.426	0.419	**0.413**	0.415	-	0.444	0.436	**0.413**	0.414	0.447	0.434	0.454	0.441	0.437	0.439	0.415	0.415	0.477	0.474
	Avg.	**0.348**	**0.381**	**0.352**	**0.378**	0.365	0.392	0.362	0.382	**0.356**	-	0.373	0.390	**0.356**	**0.381**	0.362	0.388	0.381	0.396	0.371	0.401	0.359	0.383	0.408	0.425
ETTm2	96	**0.162**	**0.253**	**0.163**	**0.253**	0.178	0.270	**0.163**	**0.252**	0.165	-	0.167	**0.250**	0.166	0.256	0.169	0.257	0.175	0.258	0.179	0.274	**0.163**	0.257	0.200	0.288
	192	**0.217**	**0.290**	**0.217**	**0.291**	0.232	0.306	**0.218**	0.294	0.219	-	0.233	**0.293**	**0.218**	**0.293**	0.231	0.299	0.237	0.299	0.239	0.314	0.223	0.304	0.274	0.337
	336	**0.267**	**0.325**	**0.269**	**0.325**	0.284	0.342	0.270	**0.327**	**0.269**	-	0.290	0.332	0.270	0.328	0.282	0.337	0.298	0.340	0.309	0.356	0.291	0.355	0.340	0.382
	720	**0.349**	**0.376**	**0.350**	**0.377**	0.360	0.389	0.352	**0.379**	**0.351**	-	0.388	0.390	0.352	0.380	0.371	0.398	0.391	0.396	0.387	0.407	0.407	0.433	0.384	0.407
	Avg.	**0.249**	**0.311**	**0.250**	**0.312**	0.264	0.327	**0.251**	**0.313**	**0.251**	-	0.270	0.316	0.252	0.314	0.263	0.323	0.275	0.323	0.279	0.338	0.271	0.337	0.300	0.354
Weather	96	**0.150**	**0.209**	**0.153**	**0.209**	0.155	0.210	0.169	0.223	0.170	-	**0.149**	**0.198**	0.174	0.230	0.160	0.213	0.163	**0.209**	0.159	0.212	0.174	0.242	0.176	0.234
	192	**0.196**	0.253	**0.199**	**0.251**	**0.199**	0.254	0.214	0.262	0.212	-	0.201	**0.244**	0.215	0.264	0.212	0.261	0.208	**0.250**	0.203	0.252	0.215	0.277	0.219	0.270
	336	**0.248**	0.289	**0.242**	**0.283**	**0.248**	0.291	0.257	0.294	0.257	-	0.259	**0.288**	0.260	0.299	0.260	0.299	0.251	**0.287**	0.253	0.291	0.262	0.319	0.277	0.311
	720	**0.315**	**0.339**	**0.314**	**0.337**	0.323	0.347	0.322	0.340	0.321	-	0.339	0.340	0.323	0.343	0.328	0.343	0.339	0.341	**0.317**	**0.337**	0.319	0.359	0.344	0.356
	Avg.	**0.227**	0.273	**0.227**	**0.270**	**0.231**	0.276	0.241	0.280	0.240	-	0.237	**0.268**	0.243	0.284	0.240	0.279	0.240	**0.272**	0.233	0.273	0.243	0.299	0.254	0.293
Electricity	96	**0.131**	**0.228**	0.136	0.232	**0.135**	**0.227**	0.139	0.234	0.138	-	0.138	0.237	0.139	**0.231**	0.143	0.244	0.153	0.247	**0.135**	0.233	0.141	0.244	0.202	0.308
	192	**0.148**	**0.244**	**0.150**	0.246	**0.149**	**0.240**	0.151	**0.245**	0.154	-	0.156	0.251	0.153	**0.245**	0.151	0.247	0.166	0.256	0.155	0.253	0.155	0.258	0.218	0.322
	336	**0.163**	**0.260**	**0.165**	0.263	**0.165**	**0.259**	0.166	**0.260**	0.170	-	0.171	0.267	0.169	0.262	0.166	0.261	0.185	0.277	0.169	0.267	0.170	0.275	0.232	0.332
	720	**0.201**	0.296	0.207	0.296	**0.202**	**0.291**	0.207	0.297	0.209	-	0.206	0.297	0.208	**0.294**	**0.194**	**0.291**	0.225	0.310	0.204	0.301	0.209	0.309	0.299	0.375
	Avg.	**0.161**	**0.257**	0.165	0.259	**0.163**	**0.254**	0.166	0.259	0.168	-	0.168	0.263	0.167	**0.258**	**0.164**	0.261	0.182	0.273	0.166	0.264	0.169	0.272	0.238	0.334
Traffic	96	**0.377**	**0.259**	0.390	**0.267**	**0.364**	**0.245**	0.389	0.268	0.389	-	0.432	0.281	0.394	**0.267**	0.398	0.297	0.462	0.285	**0.374**	0.273	0.396	0.272	0.605	0.325
	192	**0.391**	**0.271**	0.398	**0.271**	**0.380**	**0.256**	0.399	**0.270**	0.403	-	0.449	0.289	0.403	0.274	0.397	0.277	0.473	0.296	**0.393**	0.283	0.404	0.275	0.627	0.340
	336	**0.403**	0.279	0.412	**0.276**	**0.401**	**0.273**	0.411	**0.276**	0.416	-	0.465	0.295	0.417	0.281	**0.403**	0.279	0.498	0.296	0.409	0.292	0.417	0.283	0.631	0.349
	720	**0.443**	**0.285**	**0.447**	**0.296**	**0.439**	**0.283**	0.449	0.297	0.452	-	0.507	0.315	0.456	0.301	0.448	0.304	0.506	0.313	0.450	0.314	0.457	0.310	0.700	0.371
	Avg.	**0.403**	**0.274**	0.412	**0.278**	**0.396**	**0.264**	0.412	**0.278**	0.415	-	0.463	0.295	0.418	0.281	0.412	0.289	0.485	0.298	**0.407**	0.291	0.419	0.285	0.641	0.346

Table 7: Supplementary multivariate forecasting results. Avg. is computed from the four displayed results for each dataset and metric. The lowest reported value is highlighted in bold, including ties.

Dataset	H	DSR-TSF (Linear)		DSR-TSF (MLP)		SparseTSF		TimeMixer		iTransformer		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
METR-LA	96	1.044	0.674	1.024	0.663	1.056	0.667	1.051	0.657	**1.023**	**0.640**	1.096	0.685
	192	1.186	0.727	1.180	0.714	1.232	0.730	**1.163**	**0.689**	1.196	0.713	1.330	0.788
	336	1.308	0.765	**1.284**	0.773	1.356	0.772	1.304	**0.737**	1.320	0.750	1.353	0.764
	720	1.537	0.866	1.495	0.889	1.562	0.885	1.564	0.879	1.474	0.834	**1.423**	**0.800**
	Avg.	1.269	0.758	**1.246**	0.760	1.302	0.764	1.271	0.741	1.253	**0.734**	1.301	0.759
PM2.5	96	**0.430**	0.439	0.431	**0.438**	0.475	0.482	0.445	0.449	0.466	0.474	0.437	0.441
	192	**0.425**	**0.434**	**0.425**	0.437	0.458	0.465	0.479	0.483	0.505	0.511	0.454	0.458
	336	**0.415**	**0.429**	0.419	0.435	0.478	0.482	0.473	0.475	0.461	0.467	0.455	0.451
	720	0.433	0.447	**0.376**	**0.444**	0.456	0.479	0.464	0.487	0.489	0.514	0.457	0.479
	Avg.	0.426	**0.437**	**0.413**	0.439	0.467	0.477	0.465	0.474	0.480	0.492	0.451	0.457
Temp	96	0.138	0.289	**0.134**	**0.282**	0.149	0.300	0.158	0.316	0.155	0.310	0.165	0.329
	192	0.157	0.310	**0.145**	**0.300**	0.168	0.323	0.163	0.312	0.168	0.324	0.174	0.335
	336	0.179	0.338	**0.167**	**0.317**	0.220	0.373	0.249	0.421	0.232	0.395	0.246	0.417
	720	**0.364**	**0.451**	0.367	0.455	0.642	0.654	0.637	0.643	0.652	0.663	0.584	0.590
	Avg.	0.210	0.347	**0.203**	**0.339**	0.295	0.413	0.302	0.423	0.302	0.423	0.292	0.418

Table 7 shows that DSR-TSF with the MLP head achieves the lowest average MSE among the reported methods on all three datasets. On METR-LA, the best model varies across horizons and metrics, and iTransformer achieves the lowest average MAE. These results support the applicability of the channel-independent formulation to additional multivariate tasks, while showing that its benefits vary with the dataset, prediction horizon, and evaluation metric.

Table 8: Average relative MSE reduction of full DSR-TSF over ablated variants. The reduction is computed from the average MSE over four prediction horizons on ETTh1, ETTh2, ETTm1, and ETTm2. Positive values indicate that full DSR-TSF achieves lower MSE.

Ablation variant	MLP Avg. Gain	Linear Avg. Gain	Overall Avg. Gain
D2: Sparse-original	7.45%	2.24%	4.85%
D2: Kernel-matched	8.47%	2.10%	5.29%
D3: Fixed route	1.73%	1.74%	1.73%
D3: w/o spectral condition	1.96%	2.03%	2.00%
D4: w/o DSR Aggregator	2.13%	1.97%	2.05%
D4: w/o Phase Shift	1.67%	1.89%	1.78%
D4: Discrete Phase Shift	1.53%	2.09%	1.81%
D5: w/o low-rank head	2.93%	1.88%	2.41%
D5: w/o SwiGLU Residual Head	1.99%	2.04%	2.02%

D Additional ablation studies

D.1 Ablation design

This appendix provides additional controlled ablation studies to examine the structural contributions of DSR-TSF. All variants are evaluated under the same data splits, input length, prediction horizons, training objective, and metrics as the full model, with MSE and MAE reported over $H \in \{96, 192, 336, 720\}$.

We summarize each ablation by the relative MSE reduction of full DSR-TSF:

$$\Delta\mathrm{MSE} = \frac{\mathrm{MSE}_{\mathrm{Ablation}} - \mathrm{MSE}_{\mathrm{DSR}}}{\mathrm{MSE}_{\mathrm{Ablation}}} \times 100\%,$$

where a positive value indicates that DSR-TSF achieves lower MSE than the corresponding variant.

The variants are grouped into four categories: single-branch aggregation and receptive-field controls, spectrum-conditioned routing ablations, dynamic response and phase-adaptation ablations, and segment-level forecasting-head ablations. This grouping separates the effects of receptive field, routing strategy, response aggregation, and prediction-head design.

Tables 9 and 10 report the full MSE and MAE results under the MLP and Linear heads, respectively. Their Gain rows compute $\Delta\mathrm{MSE}$ from the average MSE over four horizons, and Table 8 further aggregates these gains across datasets and heads.

D.2 Single-branch aggregation baselines

We first examine whether the improvement of DSR-TSF can be attributed to single-branch aggregation or an enlarged temporal receptive field. Sparse-original follows the SparseTSF-style single-branch design, whereas DSR-TSF uses the multi-branch routed aggregation in Eq. (9). Kernel-matched further controls for receptive-field enlargement by matching the convolutional kernel size to the response branches in DSR-TSF, while retaining a single aggregation path.

DSR-TSF achieves average MSE reductions of 4.85% over Sparse-original and 5.29% over Kernel-matched. Since Kernel-matched already accounts for kernel-size enlargement, the gain cannot be sufficiently explained by a larger receptive field alone. The results instead support the benefit of spectrum-conditioned multi-branch aggregation, where different period-response bases are combined according to the current input spectrum rather than using a fixed single response path.

D.3 Routing mechanism ablation

We next evaluate whether input-dependent spectral routing remains necessary once multiple response branches are available. Fixed route replaces the routing weights in Eq. (7) with uniform branch weights, testing whether simple multi-branch averaging is sufficient. The w/o spectral condition variant removes the descriptor $f(\bar{x})$ from Eq. (7) and keeps only the learnable branch prior, testing whether global branch preferences can replace window-conditioned routing.

DSR-TSF reduces average MSE by 1.73% compared with Fixed route and by 2.00% compared with w/o spectral condition. Although these gains are smaller than those against single-branch baselines,

Table 9: Full ablation results of DSR-TSF with the MLP head. Each cell reports MSE and MAE. The Gain row reports the relative MSE reduction of full DSR-TSF over each ablated variant, computed from the average MSE over four prediction horizons.

Dataset	Horizon	DSR-TSF	Sparse	Kernel	Fixed	w/o Spec.	w/o DSR	w/o Phase	Disc. Phase	w/o LR	w/o SwiGLU
ETTh1	96	0.350/0.384	0.374/0.395	0.374/0.394	0.354/0.385	0.357/0.385	0.363/0.385	0.361/0.389	0.354/0.385	0.353/0.383	0.354/0.385
	192	0.391/0.407	0.417/0.421	0.417/0.421	0.393/0.407	0.400/0.410	0.407/0.409	0.393/0.407	0.393/0.407	0.399/0.411	0.393/0.407
	336	0.416/0.420	0.454/0.442	0.454/0.442	0.428/0.431	0.427/0.426	0.437/0.425	0.429/0.432	0.428/0.432	0.423/0.433	0.429/0.432
	720	0.415/0.445	0.487/0.485	0.487/0.485	0.432/0.448	0.424/0.447	0.425/0.443	0.430/0.450	0.430/0.450	0.442/0.460	0.430/0.450
	Avg.	0.393/0.414	0.433/0.436	0.433/0.436	0.402/0.418	0.402/0.417	0.408/0.416	0.403/0.420	0.401/0.419	0.404/0.422	0.402/0.419
	Gain	–	9.24%	9.24%	2.18%	2.24%	3.68%	2.54%	2.06%	2.78%	2.12%
ETTh2	96	0.271/0.333	0.307/0.360	0.302/0.360	0.273/0.336	0.274/0.337	0.277/0.338	0.274/0.336	0.273/0.334	0.280/0.338	0.273/0.334
	192	0.331/0.370	0.371/0.408	0.370/0.403	0.344/0.381	0.345/0.381	0.335/0.375	0.335/0.375	0.338/0.376	0.340/0.377	0.338/0.376
	336	0.351/0.392	0.405/0.435	0.409/0.436	0.355/0.394	0.355/0.395	0.352/0.392	0.358/0.396	0.358/0.396	0.365/0.403	0.358/0.396
	720	0.377/0.421	0.448/0.476	0.437/0.468	0.379/0.425	0.380/0.423	0.378/0.421	0.383/0.424	0.385/0.432	0.391/0.430	0.383/0.425
	Avg.	0.333/0.379	0.383/0.420	0.380/0.417	0.338/0.384	0.339/0.384	0.336/0.382	0.338/0.383	0.339/0.385	0.344/0.387	0.338/0.383
	Gain	–	13.13%	12.38%	1.55%	1.77%	0.89%	1.48%	1.77%	3.34%	1.63%
ETTm1	96	0.292/0.352	0.315/0.367	0.313/0.364	0.302/0.356	0.303/0.357	0.307/0.361	0.296/0.352	0.296/0.353	0.303/0.357	0.309/0.352
	192	0.333/0.372	0.347/0.388	0.341/0.382	0.337/0.375	0.338/0.376	0.341/0.378	0.338/0.376	0.338/0.376	0.343/0.383	0.347/0.378
	336	0.360/0.390	0.378/0.404	0.386/0.412	0.362/0.391	0.363/0.392	0.368/0.393	0.363/0.392	0.363/0.392	0.363/0.397	0.362/0.392
	720	0.407/0.411	0.440/0.435	0.449/0.439	0.425/0.424	0.426/0.424	0.420/0.423	0.416/0.421	0.416/0.421	0.437/0.440	0.422/0.419
	Avg.	0.348/0.381	0.370/0.399	0.372/0.399	0.356/0.387	0.358/0.387	0.359/0.389	0.353/0.385	0.353/0.386	0.362/0.394	0.360/0.385
	Gain	–	5.95%	6.51%	2.38%	2.66%	3.06%	1.49%	1.49%	3.73%	3.33%
ETTm2	96	0.162/0.253	0.177/0.263	0.177/0.263	0.164/0.254	0.166/0.255	0.165/0.256	0.164/0.254	0.164/0.254	0.163/0.254	0.164/0.254
	192	0.217/0.290	0.217/0.290	0.245/0.306	0.219/0.292	0.219/0.294	0.219/0.292	0.219/0.293	0.219/0.293	0.218/0.292	0.219/0.293
	336	0.267/0.325	0.270/0.325	0.283/0.337	0.271/0.326	0.274/0.330	0.272/0.328	0.275/0.327	0.271/0.327	0.271/0.326	0.272/0.327
	720	0.349/0.376	0.349/0.376	0.354/0.380	0.352/0.378	0.351/0.379	0.351/0.379	0.352/0.378	0.352/0.377	0.365/0.384	0.352/0.381
	Avg.	0.249/0.311	0.253/0.314	0.265/0.322	0.252/0.313	0.253/0.314	0.252/0.314	0.253/0.313	0.252/0.313	0.254/0.314	0.252/0.314
	Gain	–	1.48%	5.76%	0.80%	1.19%	0.89%	1.19%	0.80%	1.87%	0.89%

Table 10: Full ablation results of DSR-TSF with the Linear head. Each cell reports MSE and MAE. The Gain row reports the relative MSE reduction of full DSR-TSF over each ablated variant, computed from the average MSE over four prediction horizons.

Dataset	Horizon	DSR-TSF	Sparse	Kernel	Fixed	w/o Spec.	w/o DSR	w/o Phase	Disc. Phase	w/o LR	w/o SwiGLU
ETTh1	96	0.347/0.381	0.360/0.388	0.360/0.387	0.357/0.385	0.358/0.386	0.358/0.386	0.361/0.389	0.361/0.389	0.357/0.392	0.361/0.388
	192	0.387/0.404	0.393/0.408	0.393/0.408	0.400/0.410	0.392/0.407	0.392/0.407	0.391/0.410	0.393/0.412	0.397/0.412	0.392/0.411
	336	0.409/0.423	0.432/0.442	0.432/0.442	0.427/0.427	0.426/0.427	0.426/0.427	0.429/0.432	0.429/0.432	0.426/0.424	0.429/0.432
	720	0.415/0.441	0.428/0.449	0.428/0.448	0.426/0.450	0.426/0.450	0.426/0.450	0.432/0.447	0.432/0.447	0.430/0.451	0.432/0.447
	Avg.	0.390/0.412	0.403/0.422	0.403/0.421	0.403/0.418	0.401/0.418	0.401/0.418	0.403/0.420	0.404/0.420	0.403/0.420	0.404/0.420
	Gain	–	3.41%	3.41%	3.23%	2.75%	2.75%	3.41%	3.53%	3.23%	3.47%
ETTh2	96	0.273/0.334	0.279/0.339	0.277/0.338	0.275/0.336	0.280/0.340	0.275/0.336	0.275/0.336	0.279/0.337	0.280/0.338	0.276/0.335
	192	0.331/0.372	0.338/0.375	0.343/0.378	0.335/0.373	0.345/0.377	0.333/0.373	0.336/0.375	0.341/0.376	0.338/0.377	0.342/0.376
	336	0.355/0.394	0.368/0.395	0.357/0.398	0.356/0.396	0.356/0.396	0.359/0.396	0.355/0.395	0.355/0.395	0.361/0.398	0.355/0.395
	720	0.376/0.420	0.379/0.423	0.380/0.423	0.379/0.423	0.379/0.423	0.379/0.423	0.379/0.422	0.380/0.422	0.381/0.423	0.380/0.422
	Avg.	0.334/0.380	0.341/0.383	0.339/0.384	0.336/0.382	0.340/0.384	0.337/0.382	0.336/0.382	0.339/0.383	0.340/0.384	0.338/0.382
	Gain	–	2.13%	1.62%	0.74%	1.84%	0.82%	0.74%	1.48%	1.84%	1.33%
ETTm1	96	0.309/0.351	0.313/0.355	0.312/0.354	0.315/0.357	0.316/0.358	0.315/0.357	0.311/0.354	0.309/0.352	0.310/0.352	0.309/0.352
	192	0.332/0.366	0.348/0.377	0.343/0.372	0.345/0.376	0.345/0.377	0.347/0.377	0.352/0.383	0.352/0.383	0.335/0.369	0.352/0.382
	336	0.360/0.383	0.372/0.391	0.375/0.392	0.369/0.389	0.370/0.389	0.373/0.393	0.365/0.388	0.365/0.388	0.368/0.387	0.365/0.388
	720	0.406/0.410	0.418/0.414	0.417/0.412	0.414/0.414	0.414/0.414	0.419/0.417	0.411/0.413	0.411/0.413	0.415/0.412	0.411/0.413
	Avg.	0.352/0.378	0.363/0.384	0.362/0.383	0.361/0.384	0.361/0.385	0.364/0.386	0.360/0.385	0.359/0.384	0.357/0.380	0.359/0.384
	Gain	–	3.03%	2.76%	2.49%	2.63%	3.23%	2.22%	2.09%	1.47%	2.09%
ETTm2	96	0.163/0.253	0.164/0.254	0.164/0.254	0.164/0.254	0.164/0.255	0.167/0.256	0.165/0.256	0.165/0.256	0.165/0.254	0.165/0.255
	192	0.217/0.291	0.217/0.291	0.219/0.294	0.220/0.293	0.218/0.292	0.219/0.293	0.218/0.292	0.218/0.292	0.218/0.291	0.218/0.291
	336	0.269/0.325	0.271/0.326	0.270/0.326	0.269/0.326	0.273/0.329	0.272/0.329	0.272/0.327	0.272/0.327	0.270/0.326	0.272/0.328
	720	0.350/0.377	0.351/0.379	0.352/0.379	0.351/0.379	0.353/0.378	0.352/0.379	0.356/0.381	0.357/0.381	0.356/0.381	0.357/0.381
	Avg.	0.250/0.312	0.251/0.313	0.251/0.313	0.251/0.313	0.251/0.313	0.253/0.314	0.253/0.314	0.253/0.314	0.252/0.313	0.253/0.314
	Gain	–	0.40%	0.60%	0.50%	0.89%	1.09%	1.19%	1.28%	0.99%	1.28%

they remain informative because they isolate the effect of spectrum-conditioned weighting once the availability of multiple response branches is already controlled. In other words, the improvement cannot be attributed simply to replacing a single response path with a multi-branch structure. Rather, the results indicate that the benefit of DSR-TSF also depends on adapting branch weights to the spectral state of each input window. This supports the view that the routing mechanism contributes not only by increasing response diversity, but by selecting and combining responses in an input-dependent manner.

D.4 Dynamic spectral response ablation

We then study the dynamic spectral response aggregator and the phase-adaptation component. The w/o DSR Aggregator variant removes $\mathrm{DSR}(\bar{x})$ from Eq. (10), directly feeding the normalized input into the segment-level prediction head. The w/o Phase Shift variant retains the routed aggregation in Eq. (9) but replaces S_{ϕ_r} with the identity mapping. Discrete Phase Shift evaluates whether a less flexible phase adjustment is sufficient.

Removing the DSR aggregator leads to an average MSE degradation of 2.05%, showing that the response module provides useful representation adaptation before segmented forecasting. The DSR module is the component that injects spectrum-conditioned temporal responses into the sequence prior to period-based segmentation. Phase-related variants have smaller but stable effects, with DSR-TSF improving over w/o Phase Shift and Discrete Phase Shift by 1.78% and 1.81%, respectively. These results suggest that phase adaptation mainly acts as an alignment refinement on top of routed aggregation, while the larger contribution comes from the spectrum-conditioned response construction itself.

D.5 Forecasting head ablation

Finally, we evaluate the segment-level prediction head. The w/o low-rank head variant replaces the rank-constrained factorization in Eq. (17) with the full segment-level mapping in Eq. (16), testing whether a less constrained inter-segment mapping is preferable. The w/o SwiGLU residual head variant removes the residual nonlinear head design while keeping the remaining framework unchanged.

DSR-TSF improves over w/o low-rank head by 2.41% on average, indicating that the rank-constrained head is not merely a parameter-saving component. Although a full segment-level mapping has more degrees of freedom, this additional flexibility does not necessarily improve generalization and may instead weaken the structural bias imposed on inter-segment forecasting relations. The rank-constrained head can therefore be interpreted as a structured constraint on segment-level prediction, rather than as a simple low-parameter substitute for a full mapping. The 2.02% average gain over w/o SwiGLU residual head further shows that the residual nonlinear head is beneficial, but not the sole source of improvement. Taken together, these results suggest that the forecasting head contributes by keeping the segment-level mapping controlled, while the main performance gain of DSR-TSF still comes from the preceding spectrum-conditioned response adaptation.

D.6 Summary of ablation findings

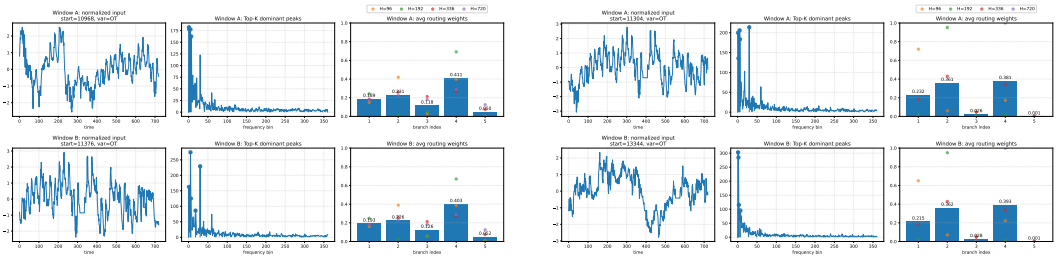
The full MSE and MAE results in Tables 9 and 10, together with the average gain summary in Table 8, support three main conclusions. First, DSR-TSF outperforms single-branch and kernel-matched variants, indicating that its improvement is not simply caused by receptive-field enlargement. Second, fixed routing and removing spectral conditioning both degrade performance, supporting the role of input-dependent spectral routing. Third, the DSR aggregator and low-rank segment-level head provide consistent structural benefits, while phase adaptation and the SwiGLU residual head act as auxiliary refinements.

Overall, these results support the main design of DSR-TSF. Spectrum-conditioned dynamic response aggregation provides the core adaptation mechanism, segmented forecasting preserves the period-aware inductive bias, and the low-rank head regularizes inter-segment prediction. The improvements are therefore better understood as the result of coordinated architectural design rather than a single isolated component.

E Mechanism analysis

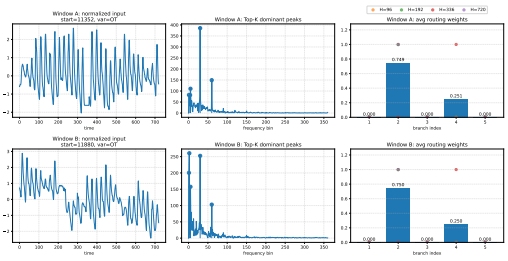
E.1 Purpose of mechanism analysis

This appendix examines whether DSR-TSF learns input-dependent routing rather than a dataset-level fixed branch mixture. We focus on three questions: whether different input windows exhibit distinct dominant spectral structures, whether the routing weights change with these spectral states, and whether the resulting routing patterns vary across forecasting horizons.

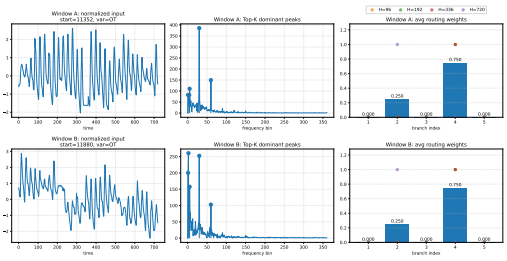


(a) ETTh1-MLP

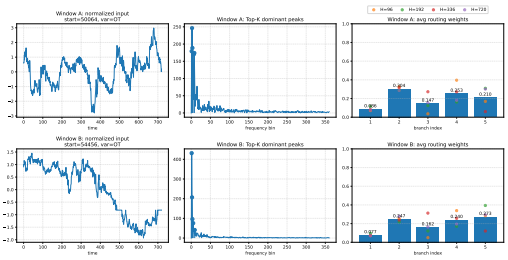
(b) ETTh1-Linear



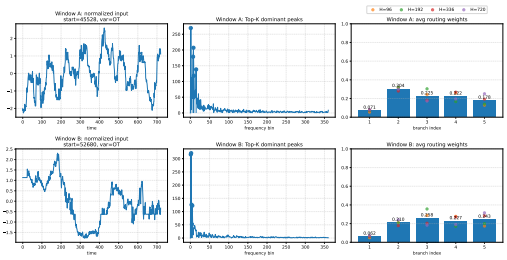
(c) ETTh2-MLP



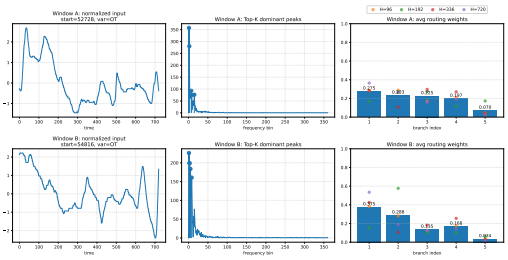
(d) ETTh2-Linear



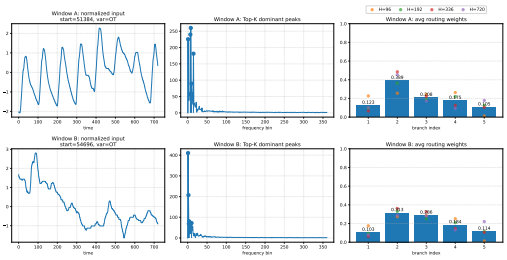
(e) ETTm1-MLP



(f) ETTm1-Linear



(g) ETTm2-MLP



(h) ETTm2-Linear

Figure 8: Spectrum–routing evidence chains across datasets and prediction heads. Each subfigure visualizes two input windows, their Top-K dominant spectral peaks, and the corresponding averaged routing weights over response branches. The results show that DSR-TSF can learn concentrated, stable, or dispersed routing patterns depending on the input spectral state, rather than using a fixed branch combination.

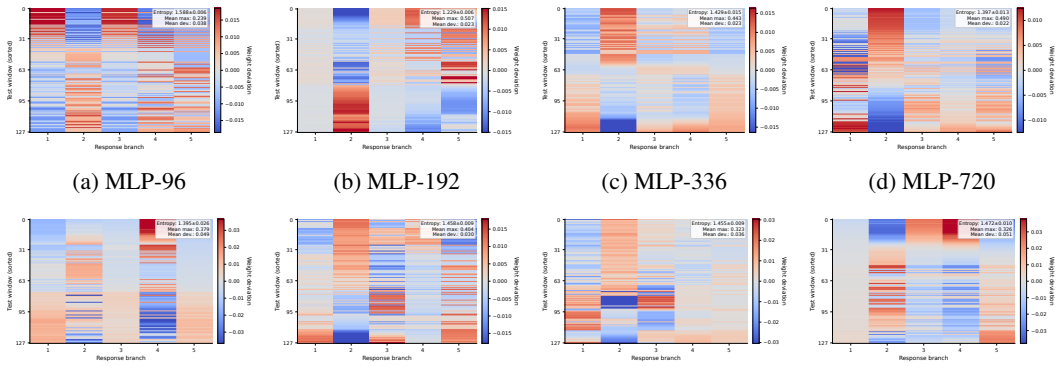
The recorded routing probabilities correspond to $w(\bar{x})$ in Eq. (7), which gives the branch allocation for each sample-channel sequence in a batch. We use these probabilities only for post-hoc visualization and summary analysis; the model architecture, training objective, and inference procedure remain unchanged. Fig. 8 provides diagnostic evidence by linking representative input windows to the Top-K dominant spectral components selected by Eq. (3) and the routing weights produced by Eq. (7). This allows us to inspect whether differences in spectral structure are reflected in branch allocation.

E.2 Spectrum-routing evidence chain

Compared with the Weather cases shown in the main text, the ETTh and ETTm examples exhibit milder window-dependent routing variation. As shown in Fig. 8, their branch distributions are often stable or only moderately dispersed across windows, indicating that DSR-TSF does not impose large routing changes when the input spectra are relatively regular. Instead, the routed aggregation remains largely stable and shifts only when the current input provides evidence for a different period-response combination.

Overall, these additional evidence chains complement the main-text Weather examples from a different perspective. While the Weather cases highlight that DSR-TSF can produce strongly input-dependent routing under more variable spectral conditions, the ETTh and ETTm cases show that the model can also maintain concentrated or mildly adaptive branch preferences when the spectral structure is more regular. This behavior is consistent with the design goal of spectrum-conditioned routing: adapting branch combinations to the observed spectral state, rather than imposing uniformly high routing variability across all datasets. These observations are also aligned with the controlled ablations in Appendix D, where removing spectral conditioning or replacing routing with fixed weights leads to consistent performance degradation.

E.3 Routing behavior across input windows



(a) MLP-96 (b) MLP-192 (c) MLP-336 (d) MLP-720

(e) Linear-96 (f) Linear-192 (g) Linear-336 (h) Linear-720

Figure 9: Routing-offset heatmaps on Weather. Rows correspond to MLP and Linear heads, and columns correspond to prediction horizons. Colors denote deviations from average branch weights.

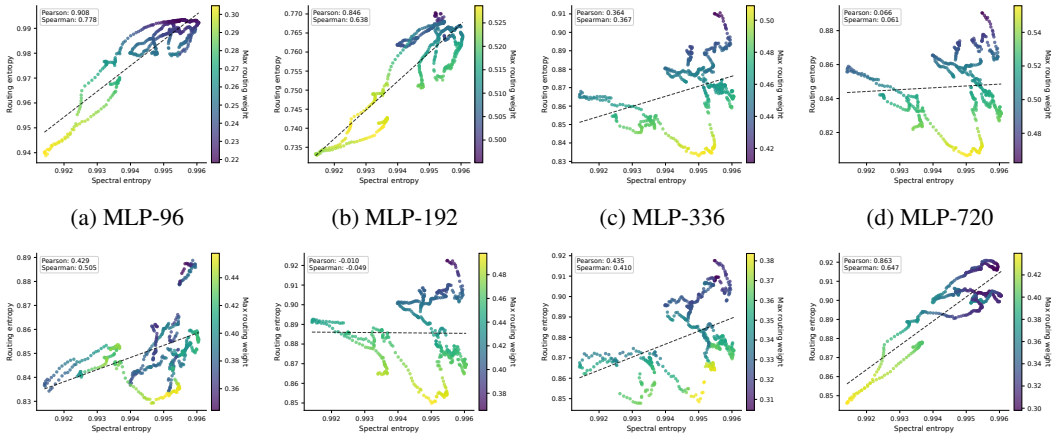
To further assess whether DSR-TSF adapts its routing across test windows, we visualize routing-offset heatmaps on Weather. Each row denotes a sorted test window, each column denotes a response branch, and the color represents the deviation of a branch weight from its average value.

Fig. 9 shows structured window-level routing variation under both prediction heads and across forecasting horizons. Some windows activate multiple branches with comparable deviations, consistent with mixed periodic evidence, whereas others concentrate the routing mass on fewer branches. Even in these more concentrated cases, the heatmaps do not collapse to a uniform pattern; instead, they retain clear positive and negative deviations across windows.

These observations indicate that dynamic routing is not expected to always produce high-entropy branch weights. Its purpose is to adjust the branch combination according to the current spectral state, balancing branch specialization and multi-branch cooperation. A fixed aggregation rule would yield nearly constant heatmaps, while the observed structured deviations show that DSR-TSF modulates branch contributions at the window level. This diagnostic evidence is consistent with the controlled ablations in Appendix D, where weakening spectrum-conditioned routing degrades performance.

E.4 Spectral entropy and routing entropy

We further examine whether the dispersion of input spectral evidence is reflected in the learned routing distribution.



(a) MLP-96 (b) MLP-192 (c) MLP-336 (d) MLP-720

(e) Linear-96 (f) Linear-192 (g) Linear-336 (h) Linear-720

Figure 10: Relation between spectral entropy and routing entropy on Weather. Rows correspond to prediction heads, and columns correspond to forecasting horizons. Spectral entropy measures the dispersion of the selected Top-K spectral evidence, while routing entropy measures the dispersion of branch allocation.

Given the Top-K amplitudes selected by Eq. (3), we compute the normalized spectral weights as

$$p_k = \frac{\log(1 + a_k)}{\sum_{i=1}^{K} \log(1 + a_i)}. \tag{19}$$

The spectral and routing entropies are defined as

$$\mathcal{H}_{\text{spec}} = -\sum_{k=1}^{K} p_k \log p_k, \qquad \mathcal{H}_{\text{route}} = -\sum_{r=1}^{R} w_r(\bar{x}) \log w_r(\bar{x}). \tag{20}$$

Larger $\mathcal{H}_{\text{spec}}$ indicates that the dominant spectral evidence is spread across more frequency components, while larger $\mathcal{H}_{\text{route}}$ corresponds to a more distributed branch allocation.

Using Eq. (20), Fig. 10 shows that the relation between spectral entropy and routing entropy varies across prediction heads and forecasting horizons. For the MLP head, the association is more evident at shorter horizons, especially $H = 96$ and $H = 192$, but weakens for longer horizons, suggesting that nonlinear long-horizon routing is also shaped by horizon-specific branch specialization. For the Linear head, positive associations appear at $H = 96$, $H = 336$, and $H = 720$, with the clearest trend at $H = 720$. The weak relation at $H = 192$ further indicates that spectral entropy is informative but not sufficient to determine the final routing distribution.

Overall, the entropy analysis provides diagnostic evidence that DSR-TSF does not rely on a dataset-level fixed branch mixture. Routing dispersion varies with the input spectral state, but is also affected by the prediction horizon and head architecture; thus, spectral entropy should be viewed as an informative factor rather than a deterministic predictor of routing behavior.

E.5 Functional analysis of response branches

We extend Fig. 5 to examine the predictive contribution and diversity of the learned response branches. On 24 frozen checkpoints, Best Single and Best Pair retain fixed subsets selected by validation MSE, while Remove-Top1 and Remove-Bottom1 discard the highest- and lowest-weight branch for each sample and channel. Retained routing weights are renormalized, allowing controlled comparisons within each trained response bank.

Using sampled test inputs, ScaleRe measures the relative squared residual after optimal scalar fitting between temporally centered branch responses. EffRank is the exponential entropy of the response matrix's normalized singular values. Scaled copies would yield ScaleRe close to zero and EffRank close to one.

Table 11: Functional analysis of response branches on ETTh1 and Weather. Test MSE is reported as mean $\pm$ sample standard deviation over three seeds; ScaleRe and EffRank report the corresponding seed averages. Bold denotes the lowest mean MSE before rounding.

Dataset	Head	H	Test MSE ↓					Response diversity	
			Full	Best Single	Best Pair	Remove-Top1	Remove-Bottom1	ScaleRe	EffRank
ETTh1	Linear	96	$\mathbf{0.361} \pm 0.017$	0.373 ± 0.016	0.367 ± 0.021	0.386 ± 0.055	0.362 ± 0.017	0.565	3.288
		720	$\mathbf{0.424} \pm 0.007$	0.594 ± 0.185	0.428 ± 0.013	1.319 ± 0.672	0.424 ± 0.007	0.300	2.587
	MLP	96	$\mathbf{0.364} \pm 0.017$	0.392 ± 0.007	0.377 ± 0.018	0.419 ± 0.071	0.366 ± 0.016	0.719	3.393
		720	$\mathbf{0.418} \pm 0.004$	0.513 ± 0.040	0.499 ± 0.035	0.463 ± 0.035	0.422 ± 0.003	0.802	4.124
Weather	Linear	96	$\mathbf{0.156} \pm 0.002$	0.243 ± 0.048	0.189 ± 0.028	0.297 ± 0.098	0.161 ± 0.006	0.614	3.032
		720	$\mathbf{0.317} \pm 0.000$	0.349 ± 0.019	0.323 ± 0.006	0.415 ± 0.050	0.322 ± 0.006	0.535	2.828
	MLP	96	$\mathbf{0.157} \pm 0.005$	0.180 ± 0.008	0.183 ± 0.018	0.193 ± 0.038	0.170 ± 0.007	0.371	2.587
		336	$\mathbf{0.250} \pm 0.001$	0.324 ± 0.024	0.260 ± 0.002	0.412 ± 0.005	0.253 ± 0.003	0.477	2.904

Table 11 shows that Full achieves lower mean MSE than both fixed subsets across all eight configurations. Removing the highest-weight branch is more harmful than removing the lowest-weight branch in all 24 checkpoints. The ScaleRe values above zero and EffRank values above one indicate multiple temporal response directions. Together, these results support functionally distinct and complementary responses with partial redundancy within the trained response bank.

E.6 Component-wise complexity breakdown

Table 12: Component-wise complexity of DSR-TSF. B, C, and L denote batch size, number of variables, and input length; K is the number of retained dominant frequencies; R is the number of response branches; h_g is the hidden size of the gating MLP; s is the response kernel size; $N = L/P$ and $M = H/P$ are the numbers of input and prediction segments; r is the factorization rank of the segment-level head in Eq. (17).

Component	Complexity	Parameters	Main role
rFFT + Top-K	$\mathcal{O}(BCL \log L)$	None	Spectral descriptor extraction
Spectral gating	$\mathcal{O}(BC(2Kh_g + h_g R))$	Small	Spectrum-conditioned routing
Response branches	$\mathcal{O}(BCRLs)$	Moderate	Multi-branch temporal response
Low-rank head	$\mathcal{O}(BCPr(N + M))$	Controlled	Segment-level forecasting

The computational overhead of DSR-TSF mainly arises from spectral conditioning and multi-branch response aggregation. For an input batch $X \in \mathbb{R}^{B \times L \times C}$, extracting the rFFT-based spectral descriptor requires $\mathcal{O}(BCL \log L)$ operations, followed by Top-K selection on the half-spectrum. This step introduces no learnable parameters, but adds a forward-pass cost for constructing the routing condition.

The gating network then maps the compact descriptor $f(\bar{x}) \in \mathbb{R}^{2K}$ in Eq. (4) to the routing weights in Eq. (7). Since this mapping is performed on the Top-K descriptor rather than the full input sequence, its cost remains limited. The main additional runtime comes from evaluating R lightweight response branches and combining them with input-dependent weights. The prediction head follows the rank-constrained factorization in Eq. (17), where the factorization rank r controls the cost of mapping $N = L/P$ historical segments to $M = H/P$ prediction segments.

Table 12 summarizes these component-wise costs. The descriptor and gating modules provide lightweight spectral conditioning, while the response branches constitute the primary extra computation introduced by DSR-TSF.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions and scope of the paper. The claims are aligned with the empirical results, ablation studies, and limitation analysis.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations and boundary conditions in the appendix B, including its scope of temporal periodic adaptation and potential failure cases under abrupt distributional changes or strong cross-variable dependencies.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Appendix A provides formal propositions on the structural properties of DSR-TSF, with explicit assumptions and complete proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the datasets, data splits, metrics, implementation details, and key hyperparameters needed to reproduce the main results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides the implementation code and reproduction instructions in the supplemental material. The code will be released publicly upon acceptance of the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the training and test settings, including data splits, evaluation metrics, optimizer, and key hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The Appendix C reports representative experiments over 10 random seeds with mean and standard deviation, and includes look-back window analysis to assess robustness across input lengths.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide sufffcient information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses potential benefits of time series forecasting and risks from erroneous downstream decisions.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or scraped datasets requiring safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper uses publicly available benchmark datasets and open-source software packages, and properly cites the original sources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The submitted code is documented with reproduction instructions and will be released publicly upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or equivalent review is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development does not involve LLMs as important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.