RaPatch: Ratio-Based Patching with Spectral Fusion for Long-Term Time Series Forecasting

Yiding Fu, Biao Luo*, Shengjie Gao
School of Computer Science and Information Engineering
Hefei University of Technology
Hefei, China
fuyiding@mail.hfut.edu.cn, luobiaohfut@163.com, gaoshengjie@mail.hfut.edu.cn

***Abstract*—Scaling patch-based forecasting to long input sequences is sensitive to the choice of patch granularity because the appropriate temporal resolution changes with both context length and forecasting horizon, making fixed patch sizes or hand-picked scales suboptimal. We propose RaPatch, a ratio-based multi-scale patching framework that learns a set of bounded patch-length ratios relative to the input length, producing multi-scale patch tokens and a ratio-based scale descriptor to condition downstream spectral modeling. By parameterizing patch lengths as ratios of the input window, the model reduces horizon-specific patch retuning and maintains stable accuracy across different look-back lengths and forecasting horizons. We further introduce MS-SFB, a frequency-domain module that aligns multi-scale spectra onto a shared frequency grid and performs ratio-conditioned filtering together with cross-variable spectral mixing to better capture periodic structure. Results on multiple public long-term forecasting benchmarks indicate that RaPatch reduces forecasting errors relative to strong patch-based and frequency-aware baselines, and the advantage is more pronounced at longer prediction horizons.**

Keywords*—*Long-term time series forecasting; Multivariate time series; Ratio-based patching; Frequency-domain modeling

I. INTRODUCTION

Time series forecasting supports planning and control in energy systems [1], transportation [2], finance [3], and healthcare [4], where accurate long-horizon predictions are often required for scheduling and resource allocation. Real-world multivariate series rarely admit a single dominant temporal scale: short-term fluctuations, seasonal cycles, and long-term trends are often entangled, and different variables may react to the same event with different delays and amplitudes under noise and non-stationarity. As the prediction horizon grows, the consequences of scale mismatch and temporal heterogeneity accumulate, making long-term time series forecasting (LTSF) challenging [5], [6], [7]. These observations motivate models that explicitly exploit multi-scale temporal structure while reducing sensitivity to horizon length and input length.

Deep learning–based LTSF models have evolved along several architectural families, including RNNs, CNNs, Transformers, MLPs [8], [9], [10], [11], and more recently state-space models (SSMs) [12]. Transformer-style architectures leverage self-attention to model long-range dependencies, while patching designs such as PatchTST [13] reduce the token length by representing the input as subseries-level patches, improving computational efficiency and encouraging local structure. Multi-scale patching frameworks like MSPatch [14] extend this idea by decomposing the series into multiple patch scales and mixing information across them. Prior analyses of patch-based models [15], [16] further suggest that the patch representation provides a strong inductive bias, sometimes accounting for a substantial portion of the observed gains.

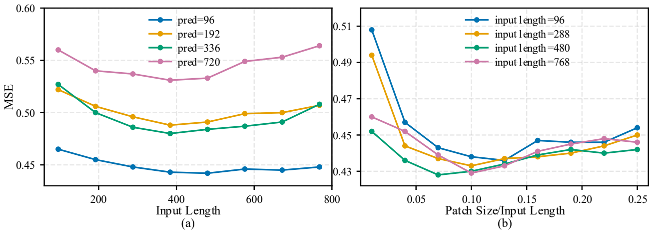

Fig. 1. Sensitivity of a patch-based Transformer to context length and patch granularity on ETTm2, a lower MSE indicates a better prediction. All experiments follow the same training protocol and model settings, varying only the indicated factor. (a) MSE versus input length L under forecasting horizons $H \in \{96,192,336,720\}$. (b) With H=720 fixed, MSE versus the patch-to-input ratio $r = P/L$ under input lengths $L \in \{96,288,480,768\}$.

Despite these benefits, patching introduces a new sensitivity: performance can vary substantially with dataset-specific settings such as the look-back window, and prediction horizon. In particular, the effective temporal granularity required for accurate forecasting changes with the horizon, so a fixed patch length can become misaligned as the horizon grows. Fig. 1(a) illustrates this dependence on the input length for a patch-based Transformer on ETTm2 under multiple horizons, and Fig. 1(b) further shows that, at a fixed long horizon H=720, forecasting error depends non-monotonically on the patch-to-input ratio across different context lengths. Moreover, existing multi-scale patching is typically defined in absolute time steps rather than as relative ratios to the input window, which makes the learned granularity harder to transfer across horizons and sampling rates.

To address these issues, we propose RaPatch, a ratio-based multi-scale patching framework for long-term multivariate time series forecasting. RaPatch parameterizes patch scale as a relative ratio of the input window and learns a small set of bounded ratios within predefined bands, yielding multi-scale patch token streams. A Band-Scale Guided (BSG) patch embedding additionally produces a compact scale descriptor that summarizes the effective ratio usage for each variable. This descriptor conditions a Multi-Source Spectral Fusion Block (MS-SFB), which aligns multi-scale patch spectra onto a shared frequency grid and performs ratio-aware dynamic band selection together with cross-variable spectral mixing in the frequency domain. Finally, a lightweight dual-branch time refiner operates in the time-domain, applying two causal depthwise-convolution branches with different receptive fields and a data-dependent gate to refine local and longer-range patterns.

Our main contributions are summarized as follows:

- We propose a Band-Scale Guided patch embedding that learns a small set of ratio-defined patch scales instead of fixed absolute patch lengths, producing multi-scale patch streams and a per-variable scale descriptor to support more robust modeling across horizons.
- We design a Multi-Source Spectral Fusion Block that uses the ratio-derived scale descriptor for dynamic frequency-band selection and cross-variable spectral mixing over multi-scale patch streams, and couple it with a Dual-branch Time Refiner based on gated short- and long-range causal convolutions, yielding an attention-free time–frequency pipeline.
- We evaluate RaPatch on multiple long-term forecasting benchmarks, comparing against patch-based Transformers, patch-based MLPs, and frequency-aware models, and observe improvements with the largest gains on the longest horizons.

II. Related Work

A. Patch-based and Multi-scale Forecasting

Recent progress in LTSF spans Transformer variants [5], [16] and strong linear or MLP baselines [11], [15], [18]. Patch-based forecasting segments long inputs into patch tokens to shorten the effective sequence length and reduce computation; PatchTST [13] shows that patch tokenization can provide a strong inductive bias with competitive accuracy. Several follow-ups strengthen patch representations via local statistics or filtering [19], [20], and Time-LLM [21] indicates that patch tokenization can also be compatible with LLM-based forecasting. Multi-scale patching further introduces multiple patch resolutions to cover heterogeneous dynamics, e.g., PatchMLP [15] and Pathformer [22], as well as MSPatch-style designs [14] that mix information across scales. However, most existing methods choose patch scales in absolute time steps and keep them fixed per dataset or horizon, which can make performance sensitive to the look-back length, and prediction horizon, increasing tuning cost and reducing transferability across settings. Moreover, multivariate series often exhibit variable-specific temporal granularities; a single global patch configuration may be suboptimal, and naively aggregating overlapping scales can lead to cross-scale misalignment when patches interleave or overlap.

B. Frequency-aware Time Series Forecasting

Frequency-domain modeling is attractive for LTSF because spectral representations can expose periodic structure and long-range regularities. Autoformer [6] uses an FFT-accelerated autocorrelation mechanism, and FEDformer [23] performs attention with DFT-based spectral representations to emphasize long-term periodic components. FreTS [24] models Fourier coefficients with frequency-domain MLPs, while FITS [25] applies rFFT and low-pass filtering to suppress high-frequency noise. TSLANet [26] introduces adaptive spectral blocks, and Affirm [27] and Filterts [28] employ learnable Fourier filtering and cross-variable filtering mechanisms to improve multivariate spectral modeling. Nevertheless, many frequency-aware designs still operate on spectra computed at a single temporal resolution and rely on frequency selection and filtering that is shared across variables or only weakly conditioned on the input window and the forecasting horizon, which can limit adaptability under horizon-dependent scale shifts and heterogeneous multivariate dynamics.

III. Method

A. Overall Architecture

Our framework consists of two core components, the Band-Scale Guided Embedding (BSG) and the Multi-Source Spectral Fusion Block (MS-SFB), as illustrated in Fig. 2. Given a multivariate input window, we first apply per-sample normalization and then use BSG to generate band-constrained multi-scale patch streams for each variable. Along with the patch tokens, BSG produces compact band–scale statistics that summarize per-variable scale usage under the learned ratio bands.

The resulting tokens are processed by a stack of encoder layers built around MS-SFB. In each layer, MS-SFB performs spectral modeling along the patch axis: it applies an FFT to obtain frequency-domain representations, aligns multi-scale spectra to a shared frequency grid, and combines ratio-conditioned band filtering, including full-band, low-pass, and high-pass filters, with energy-guided cross-variable spectral mixing. The fused spectrum is then transformed back via an inverse FFT to produce updated token representations. After the inverse FFT, a lightweight dual-branch time refiner further refines temporal patterns using gated causal depthwise convolutions with short and long receptive fields. Finally, a linear prediction head maps the encoded representations to the forecasting horizon, and the outputs are de-normalized to the original scale to produce the final forecasts.

B. Band-Scale Guided Embedding

Patch tokenization provides an efficient way to represent long inputs, but fixed patch lengths impose a single temporal granularity, and unconstrained multi-scale heuristics often require per-dataset or per-horizon tuning. In multivariate forecasting, different variables can favor different effective temporal spans, so a single global patch configuration may be misaligned with the input window length, and target horizon. The Band-Scale Guided Embedding module addresses this issue by introducing an explicit scale structure through a small set of ratio bands and by exposing per-variable scale statistics that can condition downstream modeling.

BSG defines a band set $\mathcal{B}$ of ratio intervals, where each band $b \in \mathcal{B}$ specifies an admissible range for the patch-to-input ratio. For an input window of length L, BSG learns a band-specific ratio r_b constrained to its interval, which determines an effective patch length $\tilde{s}_b \approx r_b L$. For each variable m, we treat the input as a univariate sequence $x_m \in \mathbb{R}^L$ and extract an overlapping sequence of segments of length $\tilde{s}_b$ using a fixed overlap rule. Each segment is resampled to a fixed internal length and then mapped to the embedding dimension through a shared linear projection. This yields a band-indexed token stream:

$$Z_{b,m} = (z_{b,1,m}, \dots, z_{b,L_b,m}) \in \mathbb{R}^{L_b \times d} \qquad (1)$$

where $z_{b,i,m} \in \mathbb{R}^d$ is the embedding of the i-th patch for variable m in ratio band b, L_b is the number of extracted patches for band b, and d is the embedding dimension. The band constraints reduce redundancy across scales and

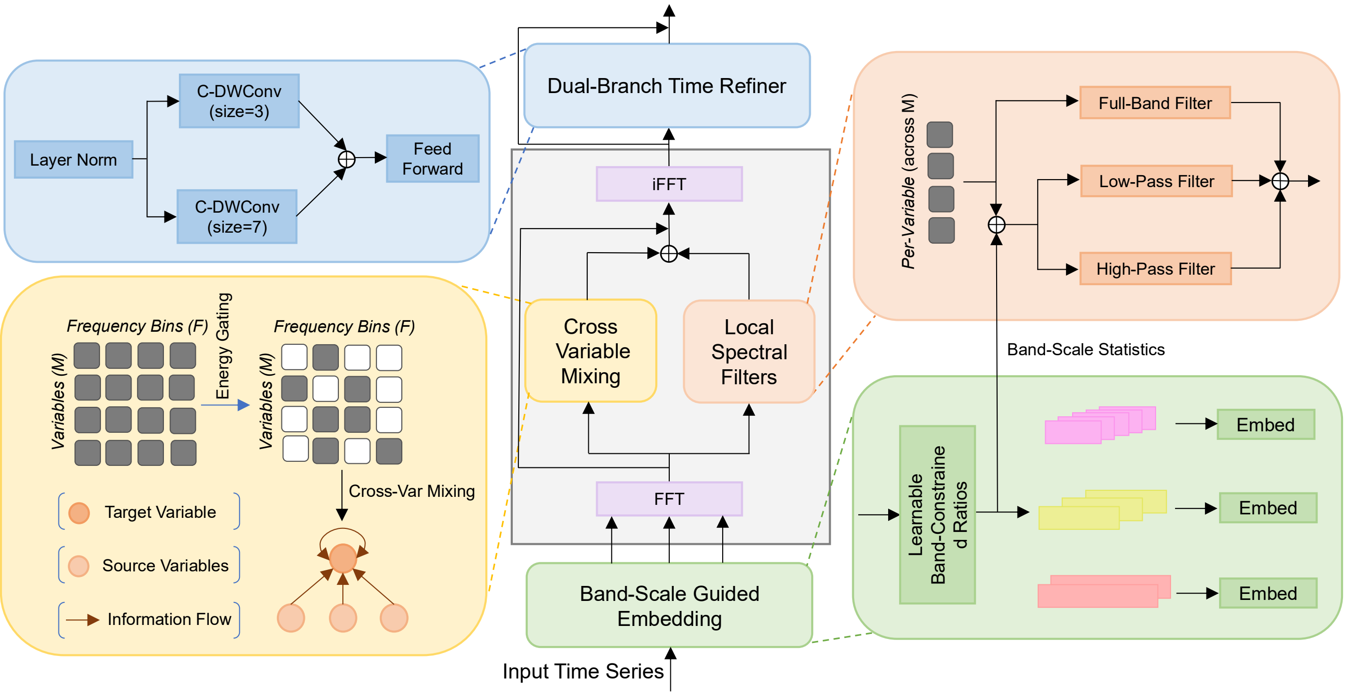

Fig. 2. Overall architecture of RaPatch. BSG normalizes the input and generates band-constrained multi-scale patch streams together with per-variable band scale statistics. Each encoder layer applies MS-SFB. MS-SFB transforms patch tokens to the frequency domain by applying an FFT along the patch axis, aligns multi-scale spectra to a shared frequency grid, and combines ratio-conditioned band filtering with energy-guided cross-variable spectral mixing before returning to the token domain via an inverse FFT. A dual-branch time refiner further enhances temporal patterns with gated causal depthwise convolutions.

encourage different bands to focus on distinct temporal spans, rather than collapsing to similar effective patch lengths.

Since patch extraction requires an integer patch length, BSG treats $\tilde{s}_b$ as a continuous value and uses a differentiable approximation during training. Specifically, it forms patch tokens using the two neighboring integer lengths and linearly interpolates the resulting token streams using the fractional part of $\tilde{s}_b$. This allows gradients to update the band ratios while keeping the patch encoder unchanged.

In addition to token streams, BSG computes per-variable band-scale statistics that summarize scale usage. For each variable $m$and ratio band b, we define an activation energy:

$$e_{b,m} = \frac{1}{L_b d} \sum_{i=1}^{L_b} \| z_{b,i,m} \|_2^2 \tag{2}$$

which measures the average squared magnitude of the patch embeddings in that band. We normalize $\{e_{b,m}\}_{b\in\mathcal{B}}$ across bands to obtain band-importance weights $w_{b,m}$, defined as $w_{b,m} = \frac{e_{b,m}}{\sum_{b'\in\mathcal{B}} e_{b',m}}, b \in \mathcal{B}$.These statistics serve as band-scale cues for the subsequent Multi-Source Spectral Fusion Block, enabling its spectral filtering and cross-variable mixing to be conditioned on the scale structure revealed at the embedding stage.

C. Multi-Source Spectral Fusion Block

MS-SFB performs spectral processing on patch-token sequences and is designed for two requirements in multivariate LTSF. First, band-scale guided tokenization produces multi-scale patch streams whose effective resolutions differ across ratio bands, variables, and samples, so spectral operations must be aligned to the resolution induced by the embedding stage. Second, variables can share coherent spectral structure, so spectral modeling should support selective cross-variable exchange rather than processing each variable in isolation. MS-SFB addresses both requirements with three stages: cross-band spectral alignment and fusion, ratio-conditioned local filtering, and dynamic cross-variable spectral fusion, followed by branch fusion and an inverse transform back to the patch domain.

1) Patch-wise Fourier Transform

BSG produces, for each variable $m$and ratio band $b \in \mathcal{B}$, a band-indexed token stream $Z_{b,m} \in \mathbb{R}^{L_b \times d}$. We treat the patch index as a one-dimensional signal and apply a real-to-complex Fourier transform along the patch axis:

$$\hat{Z}_{b,m} = \mathrm{rFFT}(Z_{b,m}) \in \mathbb{C}^{F_b \times d} \tag{3}$$

Different ratio bands use different effective patch lengths and strides, hence their spectra have different frequency resolutions. MS-SFB aligns each $\hat{Z}_{b,m}$ to a shared reference rFFT grid with $F = \lfloor N/2 \rfloor + 1$ bins, where N is the reference token length on the shared grid, set by the finest-stride ratio band, namely the band with the smallest effective stride. The reference grid is defined using the smallest stride across bands. Let $f \in \{0, \dots, F-1\}$ denote the reference frequency-bin index, and let $\nu_f = f/(F-1) \in [0,1]$ denote its normalized frequency coordinate, where $\nu_0 = 0$ is the DC component and $\nu_{F-1} = 1$ is the highest non-redundant frequency bin, corresponding to the Nyquist frequency when N is even. Each band's rFFT bins are first mapped onto ν based on its effective stride, then aligned to the reference grid via complex-valued linear interpolation. Frequencies beyond the band-specific Nyquist limit are masked to zero after interpolation. Let $\tilde{Z}_{b,m}(f) \in \mathbb{C}^d$ denote the aligned spectrum at reference bin f.We fuse ratio bands at each frequency by:

$$\hat{X}_m(f) = \sum_{b\in\mathcal{B}} \alpha_{b,m}(f) \tilde{Z}_{b,m}(f) \tag{4}$$

Here $\alpha_{b,m}(f)$ are nonnegative mixing weights, normalized over $b \in \mathcal{B}$ to sum to one for each m and f. We compute $\alpha_{b,m}(f)$ by applying a softmax over ratio bands to a magnitude-based score at frequency f. In practice, the score is the mean channel magnitude of $\tilde{Z}_{b,m}(f)$, stabilized by a $\log(1+\cdot)$ transform and scaled by a temperature parameter. This produces a unified per-variable spectrum $\hat{X}_m \in \mathbb{C}^{F\times d}$.

2) Local Frequency Filtering

MS-SFB conditions local frequency selection on the per-variable scale statistics from BSG. Using band ratios $\{r_b\}$ and band-importance weights $\{w_{b,m}\}$, we define a per-variable scale descriptor:

$$\bar{r}_m = \sum_{b\in\mathcal{B}} w_{b,m}\ r_b \in (0,1) \tag{5}$$

We map $\bar{r}_m$ to variable-specific low and high cutoffs on the normalized frequency axis and construct smooth low-pass and high-pass masks with sigmoid transitions. Let $\nu_f = \frac{f}{F-1} \in [0,1]$ be the normalized frequency coordinate of bin f. We define $\nu_m^{\text{low}} = \alpha\,\bar{r}_m$ and $\nu_m^{\text{high}} = \nu_m^{\text{low}} + \Delta$, clip ν_m^{low} and ν_m^{high} to $[0,1]$ before constructing the masks:

$$M_{\text{full},m}(f) = 1 \tag{6}$$

$$M_{\text{low},m}(f) = \sigma\left(\frac{\nu_m^{\text{low}}-\nu_f}{\tau}\right) \tag{7}$$

$$M_{\text{high},m}(f) = \sigma\left(\frac{\nu_f-\nu_m^{\text{high}}}{\tau}\right) \tag{8}$$

Here $\sigma(\cdot)$ is the sigmoid function, $\tau > 0$ controls the transition sharpness, and the masks take values in $[0,1]$. Low-pass and high-pass refer to frequency bands on the aligned spectrum and are distinct from the ratio bands used by BSG.

We compute the local spectral branch as:

$$\hat{L}_m = \sum_{k\in\{\text{full, low, high}\}} \mathcal{G}_k\left(\hat{X}_m \odot M_{k,m}\right) \tag{9}$$

Here $\hat{L}_m \in \mathbb{C}^{F\times d}$, $\odot$ denotes elementwise multiplication with the mask broadcast over the channel dimension, and $\mathcal{G}_{\text{full}}$, $\mathcal{G}_{\text{low}}$, $\mathcal{G}_{\text{high}}$ are frequency-wise complex filters applied per channel. Each complex filter is implemented as learnable complex gains per frequency and channel with a complex nonlinearity that applies ReLU to the real and imaginary parts:

$$\mathcal{G}_k(U) = (\sigma_{\mathbb{C}}(U \odot W_{k,1})) \odot W_{k,2} \tag{10}$$

$$\sigma_{\mathbb{C}}(a+ib) = \text{ReLU}(a) + i\,\text{ReLU}(b) \tag{11}$$

where $W_{k,1}, W_{k,2} \in \mathbb{C}^{F\times d}$ are learnable parameters.

3) Dynamic Cross-Variable Fusion

Local filtering refines each variable spectrum, but multivariate forecasting can benefit from routing salient spectral components across variables. Let $\hat{X} \in \mathbb{C}^{M\times F\times d}$ denote the unified spectrum after cross-band fusion, where M is the number of variables, F is the number of frequency bins on the shared grid, and d is the embedding dimension. We summarize spectral amplitudes by:

$$A_{m,f} = \frac{1}{d}\sum_{c=1}^{d} |\,\hat{X}_{m,f,c}\,|, A \in \mathbb{R}^{M\times F} \tag{12}$$

Based on A, we compute a synchronization gate $G \in [0,1]^{M\times F}$ as the product of a global frequency gate and a variable-wise gate, implemented by lightweight projections with sigmoid activation. The global gate takes the mean amplitude over variables at each frequency as input, and the variable-wise gate takes the per-variable amplitude profile as input. We apply the gate by $\hat{X}_{\text{sync}} = G \odot \hat{X}$, where $\odot$ denotes elementwise multiplication with broadcasting over the channel dimension. We then compute a soft saliency mask $H \in [0,1]^{M\times F}$ using a per-variable soft quantile threshold over frequencies, implemented as a sigmoid relaxation around the threshold with temperature t, and obtain $\hat{X}_{\text{mask}} = H \odot \hat{X}_{\text{sync}}$.

Cross-variable interactions are parameterized by a learnable mixing matrix $W \in \mathbb{R}^{M\times M}$, normalized row-wise by a softmax. The cross-variable branch is:

$$\hat{Z} = (1-\gamma)\hat{X}_{\text{sync}} + \gamma(\text{softmax}_{\text{row}}(W)\hat{X}_{\text{mask}}) \tag{13}$$

where γ controls the balance between self and cross-variable contributions. The matrix multiplication is applied over the variable dimension for each frequency bin and embedding channel.

4) Inverse Fourier Transform

MS-SFB fuses the local branch and the cross-variable branch with learnable branch weights and a residual connection to the unified spectrum. We then apply an inverse rFFT over frequency bins to reconstruct a sequence along the patch axis:

$$Y = \text{irFFT}\left(w_{\text{loc}}\hat{L}+w_{\text{rel}}\hat{Z}+\lambda\hat{X}\right) \tag{14}$$

The inverse transform is applied along the frequency axis and returns a real-valued length N sequence along the patch dimension. w_{loc} and w_{rel} are obtained by applying a softmax to learnable logits, and λ is a learnable residual scale constrained between 0 and 1. The output Y is a frequency-enhanced patch representation that is passed to the subsequent time-domain refinement block.

D. Dual-branch Time Refiner

Given the time-domain token sequence $X_t \in \mathbb{R}^{N\times d}$, we introduce a lightweight Dual-branch Time Refiner to adjust local transients and longer-range deviations in the patch domain. We first normalize features with LayerNorm $H = \text{LN}(X_t)$, and then apply two parallel causal depthwise temporal convolution branches with complementary receptive fields. The short branch uses kernel size k_s, while the long branch expands the context using a larger kernel k_l and dilation d_l:

$$U_s = \phi\left(PW_s\left(C\text{-}DWConv_{k_s}(H)\right)\right) \tag{15}$$

$$U_l = \phi\left(PW_l\left(C\text{-}DWConv_{k_l,d_l}(H)\right)\right) \tag{16}$$

Here $\text{C}-\text{DWConv}$ denotes a causal depthwise 1D convolution along the token axis, PW is a pointwise linear projection applied per token, and $\phi(\cdot)$ is a fixed nonlinearity.

TABLE I. MULTIVARIATE LONG-TERM TIME SERIES FORECASTING RESULTS ON EIGHT BENCHMARKS UNDER FOUR PREDICTION HORIZONS $H \in \{96,192,336,720\}$. WE REPORT MSE AND MAE AVERAGED OVER ALL VARIABLES AND FORECAST STEPS UNDER THE COMMON EVALUATION PROTOCOL DESCRIBED IN SECTION IV-A. LOWER IS BETTER. BOLD DENOTES THE BEST RESULT.

Models		RaPatch		TimeMixer		iTransformer		PatchTST		Crossformer		FEDformer		TimesNet		DLinear		RLinear	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.332	0.365	**0.328**	**0.361**	0.334	0.368	0.337	0.365	0.462	0.483	0.379	0.414	0.349	0.380	0.364	0.393	0.354	0.387
	192	0.381	**0.385**	**0.376**	0.388	0.377	0.391	0.379	0.390	0.528	0.519	0.437	0.442	0.397	0.409	0.414	0.423	0.405	0.417
	336	**0.418**	**0.410**	0.423	0.415	0.426	0.420	0.435	0.420	0.611	0.565	0.497	0.481	0.454	0.443	0.471	0.455	0.465	0.449
	720	**0.484**	**0.452**	0.496	0.463	0.491	0.459	0.505	0.464	0.726	0.637	0.585	0.531	0.533	0.487	0.558	0.508	0.544	0.498
	Avg	**0.404**	**0.403**	0.406	0.407	0.407	0.410	0.414	0.410	0.582	0.551	0.475	0.467	0.433	0.430	0.452	0.445	0.442	0.438
ETTm2	96	**0.168**	**0.251**	0.176	0.258	0.182	0.264	0.180	0.261	0.248	0.347	0.205	0.296	0.189	0.273	0.196	0.284	0.192	0.277
	192	**0.244**	0.308	0.246	**0.303**	0.250	0.309	0.253	0.309	0.350	0.410	0.290	0.350	0.263	0.322	0.274	0.334	0.268	0.329
	336	0.312	**0.342**	**0.307**	0.346	0.311	0.348	0.316	0.352	0.444	0.469	0.364	0.396	0.332	0.367	0.344	0.380	0.337	0.371
	720	**0.393**	**0.399**	0.422	0.427	0.412	0.407	0.427	0.413	0.612	0.566	0.492	0.472	0.445	0.435	0.469	0.450	0.453	0.441
	Avg	**0.279**	**0.325**	0.288	0.334	0.289	0.332	0.294	0.334	0.414	0.448	0.338	0.379	0.307	0.349	0.321	0.362	0.313	0.355
ETTh1	96	**0.379**	**0.401**	0.381	0.403	0.386	0.405	0.393	0.409	0.531	0.531	0.441	0.454	0.403	0.423	0.425	0.434	0.412	0.427
	192	**0.437**	0.439	0.439	**0.433**	0.441	0.436	0.451	0.440	0.622	0.580	0.507	0.494	0.467	0.455	0.486	0.474	0.474	0.464
	336	**0.469**	**0.442**	0.477	0.452	0.487	0.458	0.504	0.464	0.698	0.616	0.567	0.522	0.523	0.481	0.542	0.500	0.532	0.493
	720	**0.486**	**0.475**	0.513	0.501	0.503	0.491	0.525	0.505	0.747	0.679	0.602	0.570	0.545	0.523	0.572	0.543	0.560	0.534
	Avg	**0.442**	**0.440**	0.453	0.447	0.454	0.448	0.468	0.455	0.650	0.602	0.529	0.510	0.485	0.471	0.506	0.488	0.495	0.480
ETTh2	96	**0.295**	0.350	0.297	**0.346**	0.297	0.349	0.304	0.354	0.413	0.461	0.338	0.394	0.313	0.366	0.327	0.378	0.318	0.371
	192	**0.377**	**0.395**	0.387	0.406	0.380	0.400	0.390	0.406	0.538	0.535	0.440	0.456	0.406	0.422	0.424	0.434	0.413	0.427
	336	**0.422**	**0.426**	0.441	0.432	0.428	0.432	0.445	0.442	0.615	0.590	0.500	0.498	0.460	0.455	0.482	0.474	0.469	0.469
	720	**0.420**	**0.437**	0.449	0.453	0.427	0.445	0.449	0.460	0.638	0.623	0.511	0.521	0.468	0.479	0.489	0.493	0.479	0.490
	Avg	**0.379**	**0.402**	0.394	0.409	0.383	0.407	0.397	0.416	0.551	0.552	0.447	0.467	0.412	0.431	0.431	0.445	0.420	0.439
Electricity	96	0.147	**0.234**	**0.145**	0.237	0.148	0.240	0.150	0.239	0.203	0.312	0.167	0.267	0.154	0.250	0.159	0.253	0.157	0.250
	192	**0.159**	0.254	0.165	**0.250**	0.162	0.253	0.166	0.259	0.226	0.335	0.186	0.285	0.171	0.264	0.176	0.269	0.172	0.266
	336	**0.174**	**0.262**	0.176	0.267	0.178	0.269	0.182	0.271	0.254	0.360	0.206	0.305	0.188	0.283	0.196	0.287	0.191	0.284
	720	**0.219**	**0.309**	0.236	0.325	0.225	0.317	0.235	0.323	0.332	0.436	0.268	0.366	0.243	0.337	0.253	0.345	0.248	0.339
	Avg	**0.175**	**0.265**	0.181	0.270	0.178	0.270	0.183	0.273	0.254	0.361	0.207	0.306	0.189	0.284	0.196	0.289	0.192	0.285
Exchange	96	**0.083**	0.206	0.084	**0.202**	0.086	0.206	0.091	0.208	0.120	0.275	0.098	0.230	0.091	0.214	0.092	0.215	0.090	0.213
	192	0.178	**0.290**	**0.174**	0.295	0.177	0.299	0.179	0.303	0.253	0.405	0.204	0.338	0.187	0.312	0.191	0.315	0.188	0.312
	336	**0.322**	**0.405**	0.329	0.414	0.331	0.417	0.337	0.421	0.481	0.570	0.388	0.476	0.353	0.442	0.361	0.443	0.354	0.436
	720	**0.822**	**0.665**	0.852	0.686	0.847	0.691	0.872	0.699	1.274	0.975	1.012	0.803	0.922	0.734	0.946	0.745	0.919	0.733
	Avg	**0.351**	**0.392**	0.360	0.399	0.360	0.403	0.370	0.408	0.532	0.556	0.426	0.462	0.388	0.426	0.398	0.430	0.388	0.424
Traffic	96	0.402	0.273	0.485	0.323	**0.395**	**0.268**	0.403	0.271	0.558	0.359	0.456	0.304	0.418	0.280	0.437	0.292	0.424	0.286
	192	0.425	0.282	0.488	0.327	**0.417**	**0.276**	0.427	0.280	0.598	0.375	0.484	0.315	0.445	0.290	0.466	0.303	0.455	0.298
	336	0.443	0.288	0.507	0.330	**0.433**	0.283	0.435	**0.279**	0.631	0.392	0.512	0.327	0.464	0.301	0.489	0.313	0.478	0.309
	720	**0.463**	**0.300**	0.549	0.342	0.467	0.302	0.488	0.311	0.707	0.427	0.562	0.353	0.510	0.324	0.537	0.339	0.525	0.333
	Avg	0.433	0.286	0.507	0.331	**0.428**	**0.282**	0.438	0.285	0.624	0.388	0.504	0.325	0.459	0.299	0.482	0.312	0.471	0.307
Weather	96	**0.166**	0.213	0.169	**0.210**	0.174	0.214	0.176	0.217	0.238	0.279	0.197	0.239	0.181	0.222	0.188	0.228	0.185	0.225
	192	0.223	**0.249**	**0.219**	0.252	0.221	0.254	0.225	0.255	0.309	0.336	0.253	0.287	0.232	0.265	0.241	0.273	0.235	0.268
	336	**0.274**	**0.289**	0.284	0.299	0.278	0.296	0.285	0.300	0.393	0.398	0.323	0.336	0.295	0.310	0.306	0.320	0.301	0.315
	720	**0.353**	**0.342**	0.367	0.354	0.358	0.347	0.371	0.354	0.526	0.476	0.425	0.401	0.386	0.370	0.402	0.379	0.392	0.376
	Avg	**0.254**	**0.273**	0.260	0.279	0.258	0.278	0.264	0.282	0.367	0.372	0.300	0.316	0.274	0.292	0.284	0.300	0.278	0.296
1st Count		23	22	6	7	3	2	0	1	0	0	0	0	0	0	0	0	0	0

TABLE II. ABLATION STUDY ON FOUR REPRESENTATIVE DATASETS, ETTM1, ECL, TRAFFIC, AND WEATHER. EACH VARIANT MODIFIES ONE COMPONENT OF RAPATCH UNDER THE SAME TRAINING AND EVALUATION PROTOCOL. WE REPORT MSE AND MAE AVERAGED OVER PREDICTION HORIZONS $H \in \{96,192,336,720\}$, ALL VARIABLES, AND ALL FORECAST STEPS. LOWER IS BETTER. BSG DENOTES THE BAND-SCALE GUIDED EMBEDDING. K DENOTES THE NUMBER OF RATIO BANDS. ALIGN+FUSE DENOTES CROSS-BAND SPECTRAL ALIGNMENT AND FUSION. BAND-COND DENOTES RATIO-CONDITIONED LOCAL FILTERING. XVARMIX DENOTES CROSS-VARIABLE SPECTRAL MIXING. DTR DENOTES THE DUAL-BRANCH TIME REFINER. N/A DENOTES NOT APPLICABLE BY DESIGN.

Variant	Modules					ETTm1 avg		ECL avg		Traffic avg		Weather avg	
	BSG K	Align+Fuse	Band-Cond	XVarMix	DTR	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o BSG fixed patching	×	*N/A*	*N/A*	✓	✓	0.426	0.416	0.196	0.286	0.466	0.306	0.272	0.286
Single-scale ratio, K=1	✓ 1	*N/A*	✓	✓	✓	0.413	0.409	0.186	0.275	0.447	0.294	0.263	0.279
w/o Align+Fuse	✓ > 1	×	✓	✓	✓	0.415	0.410	0.188	0.277	0.451	0.295	0.265	0.282
w/o Band-Cond static bands	✓ > 1	✓	×	✓	✓	0.411	0.408	0.185	0.274	0.449	0.294	0.262	0.280
w/o XVarMix	✓ > 1	✓	✓	×	✓	0.410	0.407	0.194	0.282	0.458	0.300	0.263	0.281
w/o DTR	✓ > 1	✓	✓	✓	×	0.417	0.412	0.180	0.271	0.450	0.295	0.266	0.284
RaPatch, ours	✓ > 1	✓	✓	✓	✓	**0.404**	**0.403**	**0.175**	**0.265**	**0.433**	**0.286**	**0.254**	**0.273**

To adaptively balance short- versus long-range corrections at each token, we compute a token-wise gate $w \in (0,1)^{N\times 1}$ from channel-reduced statistics of H. Let $a = \text{Mean}\,(|\,H\,|) \in \mathbb{R}^N$ and $b = \text{Mean}\,(|\,H - \text{Shift}\,(H)\,|) \in \mathbb{R}^N$, where Mean (·

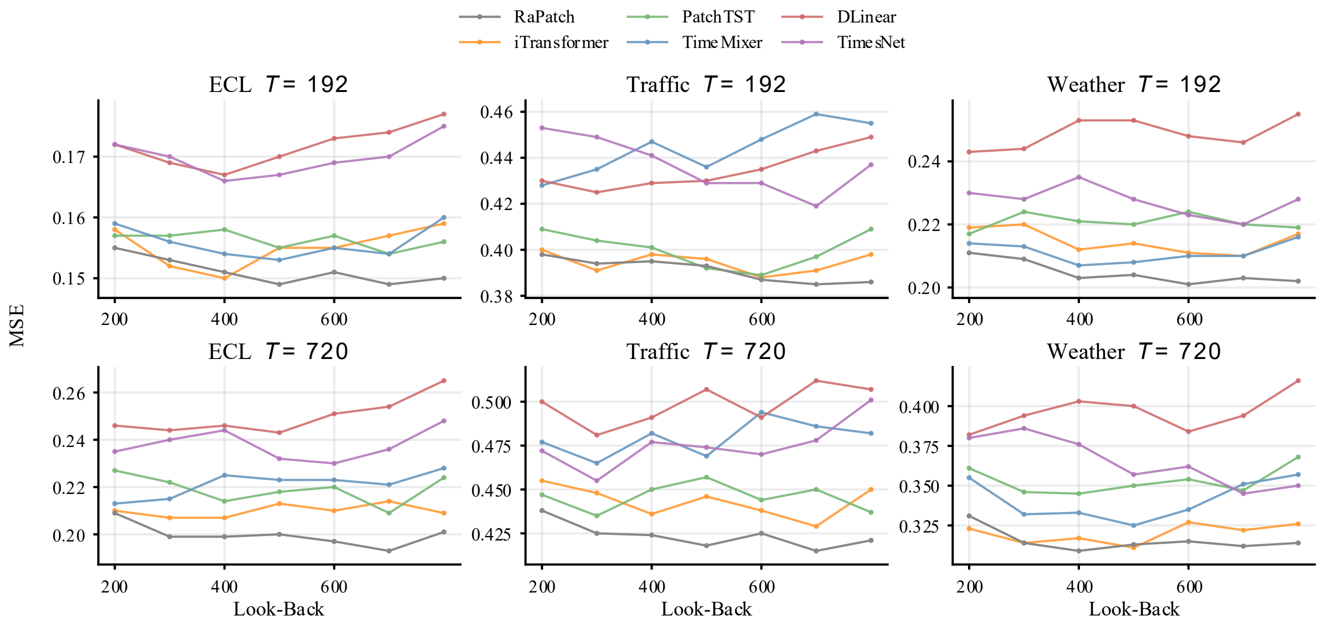

Fig. 3. Sensitivity to the look-back length L on ECL, Traffic, and Weather for horizons H=192 and H=720. The y-axis reports MSE averaged over all variables and forecast steps. All methods are evaluated under the same protocol while varying only L, lower is better.

) averages over channels and Shift $(H)_t = H_{t-1}$ is a one-step causal shift. We predict the gate by:

$$w = \sigma(g([a, b])) \tag{17}$$

where $g(\cdot)$ is a lightweight MLP applied per token and $\sigma(\cdot)$ is the sigmoid function. The two branches are fused by gated interpolation with broadcasting over channels:

$$U = w \odot U_s + (1 - w) \odot U_l \tag{18}$$

Finally, we apply a token-wise feed-forward network and LayerScale to obtain the refined features:

$$O_{\mathrm{DTR}} = \eta \odot \mathrm{FFN}\,(U) \tag{19}$$

where $\eta \in \mathbb{R}^d$ is a learnable LayerScale vector initialized to a small value.

IV. Experiments

A. Experimental Setup

1) Datasets

We evaluate our model on eight widely used long-term time series forecasting benchmarks: the four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) [5] and Electricity (ECL), Exchange-Rate (Exchange), Traffic, and Weather [6], which cover industrial processes, power systems, financial markets, road networks, and meteorology.

2) Baselines and Metrics

We compare our model with eight representative baselines, including iTransformer [10], PatchTST [13], Crossformer [29], DLinear [30], Rlinear [31], Timemixer [32], FEDformer [23], and TimesNet [33]. All methods are evaluated under the same data splits and forecasting settings, and we report MSE and MAE averaged over all variables and forecast steps.

3) Implementation Details

All experiments are conducted on an NVIDIA GeForce RTX 4090D GPU. We follow an experimental protocol similar to iTransformer [10] for fair comparison. The input length is set to 96 and we evaluate prediction horizons of 96, 192, 336, and 720. We follow iTransformer [10] for data splits and training protocol. All datasets are normalized using statistics computed from the training data. We optimize models with the mean squared error loss and select the best checkpoint based on validation performance.

B. Main Results

Table I reports results on eight benchmarks under four horizons. RaPatch achieves the best performance in 23 of 32 settings in MSE and 22 of 32 settings in MAE. This count covers eight datasets and four horizons. The gains are most pronounced at the longest horizon H=720, where RaPatch ranks first across all datasets for both metrics, covering 16 of 16 entries. On Exchange at H=720, RaPatch improves over iTransformer from MSE 0.847 and MAE 0.691 to MSE 0.822 and MAE 0.665, corresponding to 2.95% and 3.76% relative reductions. On Electricity, it improves from MSE 0.225 and MAE 0.317 to MSE 0.219 and MAE 0.309, corresponding to 2.67% and 2.52% relative reductions. At shorter horizons, some baselines such as Timemixer and iTransformer occasionally match or exceed the best score, while RaPatch remains competitive. On Traffic, RaPatch is slightly worse

than iTransformer on the average across horizons, but still marginally improves the H=720 case with MSE 0.466 and MAE 0.301 compared with MSE 0.467 and MAE 0.302. Compared with Timemixer on Traffic at H=720, RaPatch reduces MSE from 0.549 to 0.466 and reduces MAE from 0.335 to 0.301, yielding 15.1% and 10.1% relative reductions.

C. Model Analysis

1) Ablation Study

Table II summarizes ablations of RaPatch on four representative datasets, ETTm1, ECL, Traffic, and Weather, with results averaged over all prediction horizons under the same training protocol. We remove BSG by reverting to fixed

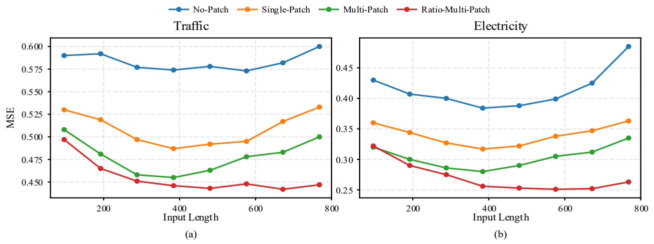

Fig. 4. Look-back scalability on Traffic and Electricity. MSE is plotted against the input length L for four patching variants: No-Patch, Single-Patch, Multi-Patch, and Ratio-Multi-Patch. All settings are kept identical except L, lower is better.

patching, reduce the ratio set to a single scale by setting K to 1, and ablate key components within MS-SFB, including spectral alignment and fusion, ratio-guided band-conditioning, and cross-variable spectral mixing, as well as the Dual-branch Time Refiner. Reverting to fixed patching yields the largest and most consistent degradation, increasing MSE by 5.4–12.0% and MAE by 3.2–7.9% across datasets. The single-scale setting, in which alignment and fusion are inactive by design, still underperforms multi-scale ratio learning, with MSE increases of 2.2–6.3%, indicating that gains come from genuine multi-scale modeling rather than patching alone. Within MS-SFB, removing alignment and fusion or replacing ratio-guided band-conditioning with static partitions consistently hurts, with MSE rising by up to 7.4% and 5.7%, respectively. Cross-variable spectral mixing is especially important on ECL and Traffic, where disabling it increases MSE by 10.9% and 5.5% and MAE by 6.4% and 4.9%. Finally, removing the Time Refiner causes a consistent 2.9–4.3% increase in MSE, showing that lightweight time-domain correction complements spectral modeling.

2) *Varying Look-back Window*

A larger look-back window increases the receptive field and provides richer historical context. We study sensitivity to the look-back length (L) on ECL, Traffic, and Weather for H=192 and H=720 by varying L while keeping other settings unchanged. For patch-based methods, we keep the patch size fixed and vary only L without retuning, to assess robustness to the look-back window. As shown in Fig. 3, increasing L does not consistently improve several baselines and can produce non-monotonic trends, especially on Traffic. In contrast, RaPatch remains competitive across different L and exhibits clearer advantages at H=720, where it achieves the best results on ECL and Traffic across all tested look-back lengths and remains close to the strongest baseline on Weather.

3) *Model Efficiency*

We examine look-back scalability on Traffic and ECL in Fig. 4. Without patching, enlarging the input window yields non-monotonic trends and may degrade forecasting quality, indicating limited ability to convert additional history into predictive context. Single-Patch reduces this sensitivity, Multi-Patch further stabilizes long-context modeling through multi-scale representations, and Ratio-Multi-Patch remains consistently strongest with the flattest trend across input lengths, suggesting more effective use of extended context with reduced reliance on carefully tuned L. We further summarize the accuracy–efficiency trade-off in Fig. 5, which plots MSE against time per iteration and uses bubble size to indicate memory footprint. Time is the average training iteration time measured under the same batch size, and memory denotes peak allocated GPU memory under the same setting. RaPatch achieves a favorable balance between accuracy, runtime, and memory compared with representative patch-based Transformer baselines and lightweight linear models.

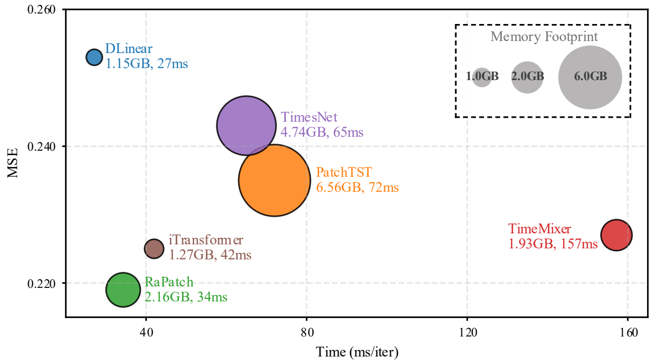

Fig. 5. Accuracy–efficiency trade-off on Electricity at H=720. The x-axis shows time per iteration (ms/iter) and the y-axis shows MSE, bubble size indicates peak GPU memory footprint. Lower and left are better.

V. Conclusion

This paper presents RaPatch, a ratio-guided multi-scale patching framework for long-term time series forecasting. RaPatch learns band-constrained patch ratios to construct ratio-defined multi-scale tokens, and combines multi-source spectral fusion with cross-variable spectral mixing and a lightweight time-domain refiner to capture periodic structure and local temporal dynamics. Across eight benchmarks, RaPatch achieves lower errors than representative baselines; the improvements are larger for long-horizon forecasts and the model is less sensitive to the chosen look-back window. Efficiency results further indicate a favorable accuracy–time–memory trade-off. Overall, ratio-guided multi-scale patching combined with spectral fusion helps capture multi-scale periodic structure, improving both accuracy and robustness for long-horizon forecasting.

References

[1] N. Ahmad, Y. Ghadi, M. Adnan, and M. Ali, “Load Forecasting Techniques for Power System: Research Challenges and Survey,” IEEE Access, vol. 10, pp. 71054–71090, 2022.

[2] A. Carianni and A. Gemma, “Overview of Traffic Flow Forecasting Techniques,” IEEE Open Journal of Intelligent Transportation Systems, vol. 6, pp. 848–882, 2025.

[3] B. H. A. Khattak, I. Shafi, A. S. Khan, E. S. Flores, R. G. Lara, and M. A. Samad, “A Systematic Survey of AI Models in Financial Market Forecasting for Profitability Analysis,” IEEE Access, vol. 11, pp. 125359–125380, 2023.

[4] M. A. Morid, O. R. L. Sheng, and J. Dunbar, “Time Series Prediction Using Deep Learning Methods in Healthcare,” ACM Trans. Manage. Inf. Syst., vol. 14, no. 1, Art. no. 2, 29 pp., 2023.

[5] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 12, 2021, pp. 11106–11115.

[6] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting,” Advances in Neural Information Processing Systems, vol. 34, pp. 22419–22430, 2021.

[7] Y. Luo, S. Zhang, Z. Lyu, and Y. Hu, “TFDNet: Time–Frequency enhanced Decomposed Network for long-term time series forecasting,” Pattern Recognition, vol. 162, Art. no. 111412, 2025.

[8] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, “SegRNN: Segment Recurrent Neural Network for Long-Term Time Series Forecasting,” IEEE Internet of Things Journal, 2025.

[9] K. Tzoumpas, A. Estrada, P. Miraglio, and P. Zambelli, “A Data Filling Methodology for Time Series Based on CNN and (Bi)LSTM Neural Networks,” IEEE Access, vol. 12, pp. 31443–31460, 2024.

[10] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, “iTransformer: Inverted Transformers Are Effective for Time Series Forecasting,” arXiv preprint arXiv:2310.06625, 2023.

[11] S.-A. Chen, C.-L. Li, N. Yoder, S. O. Arik, and T. Pfister, “TSMixer: An All-MLP Architecture for Time Series Forecasting,” arXiv preprint arXiv:2303.06053, 2023.

[12] A. Gu, K. Goel, and C. Ré, “Efficiently Modeling Long Sequences with Structured State Spaces,” arXiv preprint arXiv:2111.00396, 2021.

[13] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A Time Series is Worth 64 Words: Long-term Forecasting with Transformers,” in International Conference on Learning Representations, 2023.

[14] Y. Cao, Z. Tian, W. Guo, and X. Liu, “MSPatch: A multi-scale patch mixing framework for multivariate time series forecasting,” Expert Systems with Applications, vol. 273, Art. no. 126849, 2025.

[15] P. Tang and W. Zhang, “Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 12, 2025, pp. 12640–12648.

[16] Z. Gong, Y. Tang, and J. Liang, “PatchMixer: A Patch-Mixing Architecture for Long-Term Time Series Forecasting,” arXiv preprint arXiv:2310.00655, 2023.

[17] J. Zhu, D. Liu, H. Chen, J. Liu, and Z. Tao, “DTSFormer: Decoupled temporal-spatial diffusion transformer for enhanced long-term time series forecasting,” Knowledge-Based Systems, vol. 309, Art. no. 112828, 2025.

[18] M. M. N. Murad, M. Aktukmak, and Y. Yilmaz, “WPMixer: Efficient Multi-Resolution Mixing for Long-Term Time Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 18, 2025, pp. 19581–19588.

[19] D. Kudrat, Z. Xie, Y. Sun, T. Jia, and Q. Hu, “Patch-wise Structural Loss for Time Series Forecasting,” arXiv preprint arXiv:2503.00877, 2025.

[20] Y. Hu, G. Zhang, P. Liu, D. Lan, N. Li, D. Cheng, T. Dai, S.-T. Xia, and S. Pan, “TimeFilter: Patch-Specific Spatial-Temporal Graph Filtration for Time Series Forecasting,” arXiv preprint arXiv:2501.13041, 2025.

[21] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, “Time-LLM: Time Series Forecasting by Reprogramming Large Language Models,” in International Conference on Learning Representations, 2024.

[22] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, “Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting,” in International Conference on Learning Representations, 2024.

[23] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting,” in International Conference on Machine Learning. PMLR, 2022, pp. 27268–27286.

[24] K. Yi, Q. Zhang, W. Fan, S. Wang, P. Wang, H. He, N. An, D. Lian, L. Cao, and Z. Niu, “Frequency-domain MLPs are More Effective Learners in Time Series Forecasting,” Advances in Neural Information Processing Systems, vol. 36, pp. 76656–76679, 2023.

[25] Z. Xu, A. Zeng, and Q. Xu, “FITS: Modeling Time Series with 10k Parameters,” arXiv preprint arXiv:2307.03756, 2023.

[26] E. Eldele, M. Ragab, Z. Chen, M. Wu, and X. Li, “TSLANet: Rethinking Transformers for Time Series Representation Learning,” arXiv preprint arXiv:2404.08472, 2024.

[27] Y. Wu, X. Meng, H. Hu, J. Zhang, Y. Dong, and D. Lu, “Affirm: Interactive Mamba with Adaptive Fourier Filters for Long-term Time Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 20, 2025, pp. 21599–21607.

[28] Y. Wang, Y. Liu, X. Duan, and K. Wang, “FilterTS: Comprehensive Frequency Filtering for Multivariate Time Series Forecasting,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 20, 2025, pp. 21375–21383.

[29] Y. Zhang and J. Yan, “Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting,” in International Conference on Learning Representations, 2023.

[30] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are Transformers Effective for Time Series Forecasting?” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 9, 2023, pp. 11121–11128.

[31] Z. Li, S. Qi, Y. Li, and Z. Xu, “Revisiting Long-term Time Series Forecasting: An Investigation on Linear Mapping,” arXiv preprint arXiv:2305.10721, 2023.

[32] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, “TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting,” arXiv preprint arXiv:2405.14616, 2024.

[33] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, “TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis,” arXiv preprint arXiv:2210.02186, 2022.