TimeMosaic: Temporal Heterogeneity Guided Time Series Forecasting via Adaptive Granularity Patch and Segment-wise Decoding

Kuiye Ding[1], Fanda Fan[1†], Chunyi Hou[2], Zheya Wang[3], Lei Wang[1], Zhengxin Yang[1], Jianfeng Zhan[1,4]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]School of Information Science and Technology, Beijing University of Technology
[3]Department of Mathematical Sciences, Durham University
[4]University of Chinese Academy of Sciences
{dingkuiye, fanfanda, wanglei_2011, yangzhengxin, zhanjianfeng}@ict.ac.cn,
houchunyi@emails.bjut.edu.cn, zheya.wang@durham.ac.uk

Abstract

Multivariate time series forecasting is essential in domains such as finance, transportation, climate, and energy. However, existing patch-based methods typically adopt fixed-length segmentation, overlooking the heterogeneity of local temporal dynamics and the decoding heterogeneity of forecasting. Such designs lose details in information-dense regions, introduce redundancy in stable segments, and fail to capture the distinct complexities of short-term and long-term horizons. We propose **TimeMosaic**, a forecasting framework that aims to address temporal heterogeneity. TimeMosaic employs adaptive patch embedding to dynamically adjust granularity according to local information density, balancing motif reuse with structural clarity while preserving temporal continuity. In addition, it introduces segment-wise decoding that treats each prediction horizon as a related subtask and adapts to horizon-specific difficulty and information requirements, rather than applying a single uniform decoder. Extensive evaluations on benchmark datasets demonstrate that TimeMosaic delivers consistent improvements over existing methods, and our model trained on the large-scale corpus with 321 billion observations achieves performance competitive with state-of-the-art TSFMs.

Code — https://github.com/BenchCouncil/TimeMosaic

1 Instruction

Multivariate time series forecasting plays a critical role in numerous real-world domains (Qiu et al. 2025a,c), such as finance (Hu et al. 2025a), transportation, climate, healthcare, and energy management. These applications require accurate and efficient modeling of intricate temporal relationships among multiple correlated variables. Recently, forecasting methods employing patch-based representations have shown remarkable performance and have become increasingly prevalent (Chen et al. 2024; Jin et al. 2024; Kudrat et al. 2025; Nie et al. 2023). Patch-based approaches excel at capturing localized temporal structures and mitigating noise through segmented encoding. Recently proposed time series models, such as Sundial (Liu et al. 2025), also use patch-based time series segmentation to organize event contexts for benchmarking time series foundation models. Typically, these methods divide input sequences into fixed-length patches, implicitly assuming uniform information density and temporal complexity across the entire sequence.

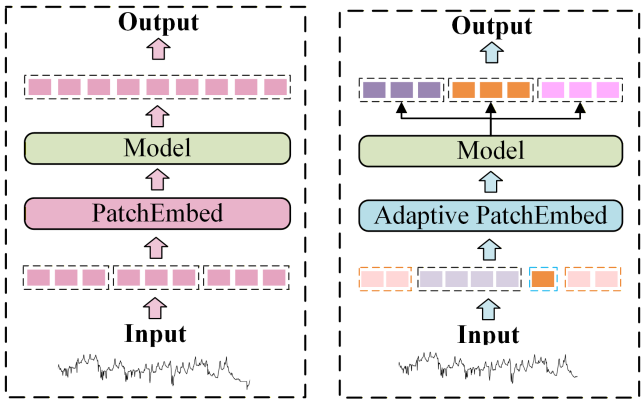

Figure 1: Comparison between existing fixed-patch models and our adaptive patching design, highlighting differences in input representation and prediction structure.

However, real-world time series often exhibit significant variability in local information density (Huang et al. 2023; Warren Liao 2005). Segments characterized by complex or abrupt changes inherently contain higher information densities, whereas smoother or stationary regions possess relatively lower densities. Current methods employing fixed-size temporal segments disregard this inherent variability, resulting in inadequate modeling of information-rich regions and redundant encoding in smoother ones. Beyond local variability, empirical evidence (Xie et al. 2025) also reveals two key structural properties: *motif reuse*, characterized by Zipf-like frequency distributions, and *structural clarity*, measured by latent-space separability (see Section 2 for definitions and illustrations), which further highlight a structural limitation of fixed-length patching: it cannot simultaneously preserve reusable long-range patterns and

†Corresponding author: fanfanda@ict.ac.cn.

maintain well-delimited local boundaries. A similar limitation has been observed in autoregressive image generation, such as DQ-VAE (Huang et al. 2023), where fixed-size region encodings likewise fail to adapt to diverse structural patterns.

These observations indicate that the limitations of fixed-length patching are not incidental but stem from a deeper property of time series data: **heterogeneity**. On the input side, local temporal regions vary greatly in complexity and information density, which we refer to as *encoding heterogeneity*. On the output side, forecasting horizons differ in both difficulty and information requirements, which we refer to as *decoding heterogeneity*. Addressing these two dimensions of heterogeneity is crucial for advancing time series forecasting.

Overall, multivariate time series forecasting faces two fundamental challenges: ① **Encoding heterogeneity.** Fixed-length segmentation fails to adapt to the variability of local temporal complexity, leading to a trade-off between reusing long-term motifs and preserving clear structural boundaries. ② **Decoding heterogeneity.** Forecasting demands differ substantially across horizons: short-term forecasting mainly rely on recent local information, while long-term forecasting require modeling more abstract and uncertain dynamics over broader contexts. Existing methods typically overlook this asymmetry by applying a single decoder to all horizons.

To address these challenges, we propose **TimeMosaic**, a unified forecasting framework that integrates adaptive patch embedding and segment-wise decoding. On the input side, it employs an adaptive granularity patching strategy inspired by dynamic quantization (Huang et al. 2023), segmenting sequences into variable-length patches according to local temporal information density to balance motif reuse and structural clarity while preserving strict temporal continuity. On the output side, a segment-wise prediction module based on multi-task prompt tuning (Liu et al. 2021; Crawshaw 2020a) treats different horizons as related subtasks, using horizon-aware prompts to capture segment-specific difficulty without altering the backbone. As shown in Figure 1, this design enables TimeMosaic to adaptively capture temporal heterogeneity and improve forecasting precision.

We summarized our main contributions as follows:

- We propose **TimeMosaic**, a novel forecasting framework that explicitly addresses both encoding heterogeneity and decoding heterogeneity in multivariate time series.
- We design an Adaptive Patch Embedding module that dynamically allocates region-specific patch sizes according to local information density, effectively balancing motif reuse with structural clarity.
- We introduce a segment-wise prompt tuning strategy that models horizon-specific difficulty and information requirements by treating each prediction segment as a distinct subtask within a unified multi-task framework.

2 Related Work

Patch-based Time Series Forecasting. Patch-based models such as PatchTST (Nie et al. 2023), Patchwise (Kudrat et al. 2025), TimeFilter (Hu et al. 2025b), and Time-LLM (Jin et al. 2024) typically use fixed-length patches, implicitly assuming uniform temporal complexity. Multi-granularity methods like PatchMLP (Tang and Zhang 2025), PathFormer (Chen et al. 2024), and DualSG (Ding et al. 2025) allow variable patching but often disturb temporal consistency due to overlapping or misordered segments (Appendix Fig. 7). In contrast, our approach adapts patch sizes to local information density while preserving strict chronological order.

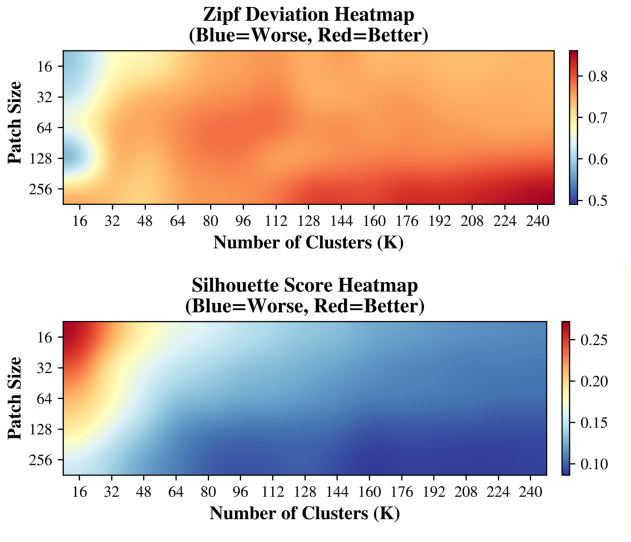

Figure 2: Zipf deviation and Silhouette score. These results are obtained by extracting patches of different lengths from a large-scale collection of time series forecasting datasets (see Appendix A), followed by K-Means clustering under various cluster settings.

Zipf Conformity and Clustering Clarity. Temporal patches exhibit Zipf-like frequency distributions (Xie et al. 2025; Axtell 2001) and clear clustering structures (Rousseeuw 1987). Zipf conformity reveals reusable motifs for compact representation, whereas clustering clarity measures pattern separability in latent space. Fixed-length patching cannot balance both: larger patches enhance motif reuse but blur boundaries, while smaller ones sharpen separation yet fragment long-term patterns, as shown in Figure 2[1]. Our adaptive patching reconciles this trade-off by dynamically allocating granularity based on local density.

Multi-task Learning and Prompt-based Adaptation. MTL (Ruder 2017; Crawshaw 2020b) enhances generalization via shared representations, while prompt-based adaptation (Lester, Al-Rfou, and Constant 2021; Xia et al. 2024) has shown strong efficiency in LLMs (Laban et al. 2023; Zhang et al. 2024; Cao et al. 2024). We extend this paradigm to time series forecasting with horizon-aware prompts that flexibly adapt to segment-specific patterns, unlike static basis-reconstruction methods (Huang et al. 2025).

[1]Appendices are available at https://arxiv.org/abs/2509.19406

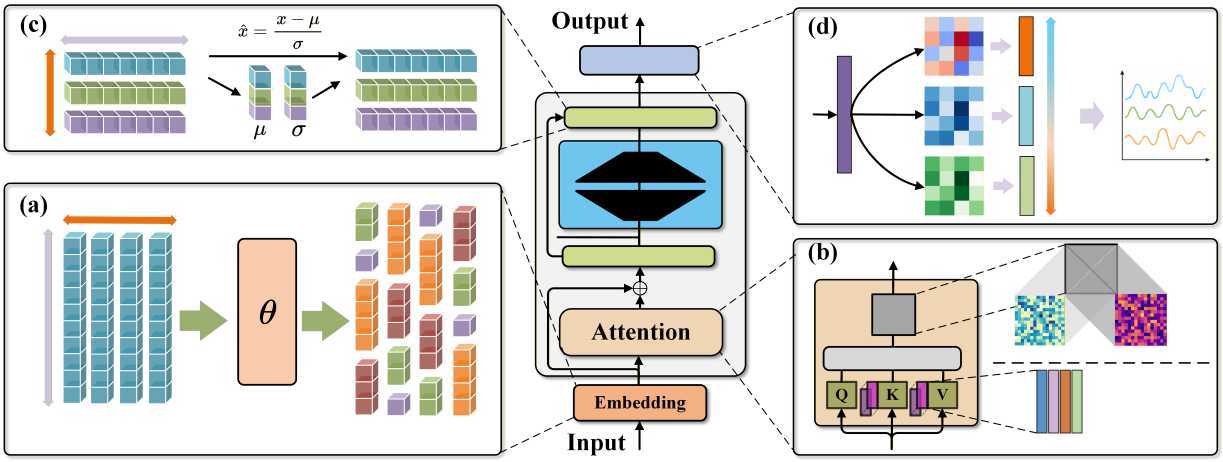

Figure 3: Overall architecture of TimeMosaic. (a) The input multivariate series is segmented by the Adaptive Patch Embedding module into variable-granularity patches via learnable region-aware decisions. (b) Learnable prompt tokens are injected and interact with patch embeddings through multi-head attention, as illustrated by the attention maps. (c) Channel-level normalization follows iTransformer (Liu et al. 2024). (d) Segment-wise forecasting is performed from short- to long-term.

3 Proposed Method

3.1 Adaptive Patch Embedding

To effectively model temporal heterogeneity in time series, we introduce an Adaptive Patch Embedding (APE) module that adjusts patch granularity based on local dynamics.

Patch Granularity Search Space. We refer to *regions* as local partitions of the input used for adaptive patching, and *segments* as forecast sub-horizons used in multi-task prediction. This distinction allows us to separately model spatial encoding and temporal decoding. Given an input time series $x \in \mathbb{R}^{B\times C\times L}$, we divide the sequence into $R = L/f_{\max}$ non-overlapping regions, where $f_{\max}$ is the maximum patch length in the candidate set:

$$\mathcal{F} = \{f_1, f_2, \ldots, f_K\}, \quad f_1 < f_2 < \cdots < f_K. \tag{1}$$

Each candidate patch size f_k represents a different temporal granularity. For each region, our goal is to select the most suitable f_k to match its local information density.

Region-wise Granularity Classification. To predict the optimal patch size for each region, we employ a lightweight classifier $\mathcal{G}_\theta$:

$$\mathbf{g}_r = \mathcal{G}_\theta(x_r) \in \mathbb{R}^K, \quad \theta_r = \arg\max_k g_{r,k}, \tag{2}$$

where g_r is the predicted logits over candidate patch sizes for region x_r, and θ_r is the index of the selected patch size. The classifier is shared across all regions and channels, and is implemented as a two-layer MLP. To ensure end-to-end differentiability, we apply the Gumbel-Softmax (Jang, Gu, and Poole 2017) during training.

Patch Alignment and Embedding. After determining the patch size f_{θ_r} for region x_r, we unfold it into patches of size f_{θ_r}:

$$z_r = \text{Linear}_{f_{\theta_r}}\left(\text{Unfold}(x_r; f_{\theta_r})\right) \in \mathbb{R}^{N_r\times d}, \tag{3}$$

where z_r is the embedded patch sequence, $N_r = f_{\max}/f_{\theta_r}$ is the number of patches in region r, and d is the embedding dimension. Each patch is projected to a shared latent dimension d using a dedicated linear layer.

To maintain a uniform input shape across regions, we upsample all patch sequences to a fixed length $N = f_{\max}/f_{\min}$ using replication:

$$\tilde{z}_r = \text{RepeatPad}(z_r; N). \tag{4}$$

where $\tilde{z}_r$ is the length-aligned patch sequence for region r, padded to $N = f_{\max}/f_{\min}$ using replication. RepeatPad preserves temporal alignment and avoids introducing artificial content. In practice, we find it more stable than interpolation or learned resampling under varying patch configurations.

All region patch embeddings $\{\tilde{z}_1, \ldots, \tilde{z}_R\}$ are concatenated and enriched with positional encodings to form the final input:

$$Z = \text{PE}\left(\text{Concat}(\tilde{z}_1, \ldots, \tilde{z}_R)\right) \in \mathbb{R}^{B\cdot C\times N\cdot R\times d}. \tag{5}$$

where Z is the final encoder input sequence after adding positional encodings, $\text{PE}(\cdot)$ denotes the positional encoding function, and R is the number of input regions. We explore several alternative designs for positional encoding, including fixed, learnable, and hybrid schemes, which are analyzed in Section 4.

Regularization via Budget Loss Without constraints, the classifier may degenerate to always selecting the finest patch size. To avoid this, we introduce a *budget loss* to match the empirical distribution of selected granularities to a predefined target:

$$\mathcal{L}_{\text{budget}} = \sum_{k=1}^{K-1} \left(r_k - \hat{r}_k\right)^2, \tag{6}$$

where r_k is the desired usage ratio of patch size f_k, and $\hat{r}_k$ is the actual usage in the current batch. The last ratio is defined by normalization:

$$r_K = 1 - \sum_{k=1}^{K-1} r_k, \quad \hat{r}_K = 1 - \sum_{k=1}^{K-1} \hat{r}_k. \tag{7}$$

Training Objective. We jointly optimize the forecasting loss and the budget regularization. The total training loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{forecast}} + \lambda \mathcal{L}_{\text{budget}}, \tag{8}$$

where $\mathcal{L}_{\text{forecast}}$ is the standard mean squared error (MSE) between the prediction $\hat{y}$ and ground truth y:

$$\mathcal{L}_{\text{forecast}} = \frac{1}{B} \sum_{i=1}^{B} \|\hat{y}^{(i)} - y^{(i)}\|_2^2. \tag{9}$$

λ is a hyperparameter balancing the two terms.

3.2 Prompt Tuning for Segment-wise Forecasting

We formulate multi-interval time series forecasting as a segment-wise multitask learning problem, where each forecast segment corresponds to an individual prediction subtask. To achieve parameter-efficient segment adaptation, we draw inspiration from Prompt Tuning (Lester, Al-Rfou, and Constant 2021; Liu et al. 2021) and extend it to the temporal domain. Specifically, we design learnable segment-aware prompts to inject segment-specific inductive biases into a shared encoder, allowing the model to specialize its attention behavior for each segment without modifying backbone parameters.

Following standard practices in prompt tuning, these prompts are inserted only into the key and value paths of the attention mechanism, while the query vectors are derived exclusively from the input data. This asymmetric design ensures that the original data tokens retain their semantic role as information seekers, while prompts act as soft guidance to steer the attention focus in a segment-aware manner. Compared with adapter-based or decoder-tuning approaches, prompt tuning achieves similar segment-level specialization with fewer trainable parameters and better modularity, as prompts can be flexibly added or removed without altering the shared encoder.

For a given segment $k \in \{1, \ldots, K\}$, we associate a prompt embedding $\boldsymbol{\phi}_k \in \mathbb{R}^{l \times d}$ and prepend it to the input representation $\mathbf{X} \in \mathbb{R}^{n \times d}$:

$$\tilde{\mathbf{X}}_k = \text{Concat}(\boldsymbol{\phi}_k, \mathbf{X}). \tag{10}$$

To preserve a clear functional separation between prompts and data tokens, we apply a prompt-masked attention mechanism. In the self-attention computation, query vectors are derived exclusively from data tokens, while key and value vectors include both data and prompts:

$$\mathbf{Q}_k = \mathbf{X}W^Q, \quad \mathbf{K}_k = \tilde{\mathbf{X}}_k W^K, \quad \mathbf{V}_k = \tilde{\mathbf{X}}_k W^V. \tag{11}$$

This allows the segment-aware prompt to modulate attention flow and inject task-relevant semantics, while being excluded from explicit decoding.

After encoding, each segment is decoded by a segment-specific head $f_k(\cdot; \theta_k)$, yielding the prediction $\hat{\mathbf{Y}}^{(k)} \in \mathbb{R}^{m_k \times C}$:

$$\hat{\mathbf{Y}}^{(k)} = f_k(\mathbf{H}_k; \theta_k), \tag{12}$$

where $\mathbf{H}_k$ is the shared encoder output.

This segment-wise prompt tuning framework enables flexible modeling of segment-specific dynamics, improves parameter efficiency, and avoids interference across subtasks. All encoder parameters remain frozen during training, and only prompts $\{\boldsymbol{\phi}_k\}$ and decoding heads $\{\theta_k\}$ are updated.

4 Experiments

We thoroughly evaluate the proposed TimeMosaic on various time series forecasting applications, validate the generality of the proposed framework and further delve into the effectiveness of adaptive patch embedding and Segment-wise Forecasting in other models.

Datasets. We conduct extensive experiments on 17 real-world multivariate time series datasets, which are divided into long-term and short-term forecasting tasks. For long-term forecasting, we use ETTh1, ETTh2, ETTm1, ETTm2 (Zhou et al. 2021), Weather, Traffic, Electricity, ExchangeRate (Wu et al. 2023), Solar-Energy (Lai et al. 2018), and Wind (Location1–4) (Xie et al. 2025). For short-term forecasting, we adopt four benchmark traffic datasets: PEMS03, PEMS04, PEMS07, and PEMS08 (Wang et al. 2024, 2025b).

Baselines. We choose the last state-of-the-art LTSF models, including TimeFilter (Hu et al. 2025b), SimpleTM (Chen et al. 2025), PatchMLP (Tang and Zhang 2025), xPatch (Stitsyuk and Choi 2025), DUET (Qiu et al. 2025b), PathFormer (Chen et al. 2024), iTransformer (Liu et al. 2024), TimeMixer (Wang et al. 2024), PatchTST (Nie et al. 2023), FreTS (Yi et al. 2023), DLinear (Zeng et al. 2023), LightTS (Zhang et al. 2022) as baselines for our experiments. We compared the ability of zero-shot with the GPT4TS (Zhou et al. 2023) and LLMTime (Nate Gruver and Wilson 2023).And we compare against pre-trained foundation time series models, including TimeMoE (Shi et al. 2025), MOIRAI (Woo et al. 2024), Chronos (Ansari et al. 2024), TimesFM (Das et al. 2024), and Moment (Goswami et al. 2024), using the parameter settings as specified in their respective publications.

Evaluation Metrics. Following previous works, we use Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics to assess the performance.

Channel Modeling Strategy. Although our core design centers on segment-wise forecasting with adaptive patch granularity, handling multivariate inputs remains essential. To ensure compatibility, we incorporate a modular *channel modeling component*, which supports both Channel-Independent (CI) (Nie et al. 2023) and Channel-Dependent (CD) (Hu et al. 2025b) schemes. Specifically, we provide multiple instantiations ranging from simple per-variable modeling to prompt-augmented or calendar-aware variants.

Models	**TimeMosaic (Ours)**		SimpleTM (2025)		TimeFilter (2025)		xPatch (2025)		PatchMLP (2025)		DUET (2025)		PathFormer (2024)		iTransformer (2024)		TimeMixer (2024)		PatchTST (2023)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.381**	**0.381**	0.391	0.403	0.395	0.407	0.391	0.402	0.401	0.407	0.416	0.407	0.392	0.385	0.409	0.411	0.381	0.396	0.387	0.398	0.407	0.409
ETTm2	0.273	**0.314**	0.283	0.325	0.282	0.329	0.283	0.327	0.287	0.331	0.286	0.328	**0.269**	0.314	0.293	0.334	0.279	0.325	0.279	0.323	0.350	0.400
ETTh1	**0.425**	**0.424**	0.442	0.436	0.463	0.447	0.446	0.439	0.455	0.445	0.447	0.440	0.454	0.428	0.450	0.442	0.445	0.439	0.432	0.431	0.462	0.459
ETTh2	**0.363**	**0.388**	0.424	0.413	0.389	0.422	0.409	0.405	0.402	0.416	0.379	0.400	0.372	0.397	0.386	0.407	0.381	0.404	0.376	0.402	0.556	0.516
Weather	0.251	0.267	0.258	0.282	0.242	0.273	0.264	0.284	0.254	0.289	0.269	0.304	**0.242**	**0.266**	0.346	0.276	0.309	0.360	0.264	0.283	0.265	0.317
Traffic	0.458	**0.283**	0.545	0.347	0.461	0.309	0.520	0.335	0.495	0.337	0.628	0.392	0.581	0.343	**0.453**	0.303	0.499	0.291	0.486	0.314	0.627	0.387
Electricity	0.187	0.279	0.248	0.284	**0.170**	**0.266**	0.209	0.292	0.224	0.289	0.216	0.311	0.214	0.293	0.225	0.310	0.185	0.294	0.205	0.292	0.215	0.305
Exchange	**0.348**	0.403	0.356	0.411	0.366	0.411	0.372	0.410	0.374	0.412	0.356	0.400	0.375	0.411	0.366	0.414	0.385	0.423	0.359	**0.401**	0.354	0.420
Solar	**0.240**	0.270	0.315	0.333	0.248	0.270	0.248	0.276	0.255	0.280	0.349	0.350	0.255	**0.258**	0.378	0.350	0.292	0.281	0.326	0.397	0.256	0.304

Table 1: Averaged forecasting results under unified experimental settings. For **long-term** forecasting tasks, we use a fixed lookback window of $L = 96$. Long-term forecasting results are averaged over four prediction lengths: $T = \{96, 192, 336, 720\}$. The best model is in **boldface**, and the second best is underlined. See Table 20 in Appendix G for complete results.

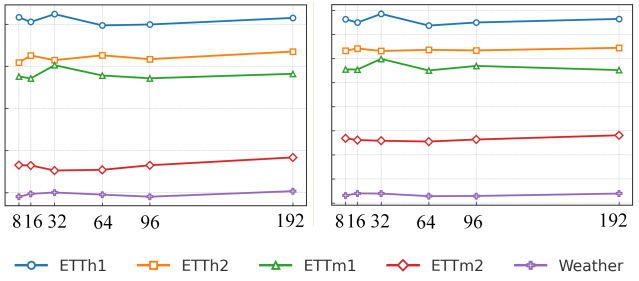

Figure 4: Segments of different sizes on five datasets with a prediction length of 192. Left: MSE. Right: MAE.

These options are fully pluggable and do not interfere with the main forecasting pipeline. By default, we adopt the Channel dependent strategy. A comprehensive description and comparison of all variants are provided in Appendix D.

Implementation Details. All the experiments are implemented in PyTorch (Paszke et al. 2019), and conducted on eight A800 GPU. Even in recent studies, it is common to set $drop\_last = True$ during data preprocessing (Chen et al. 2025; Wang et al. 2025a), and this issue has been explicitly discussed in the TFB (Qiu et al. 2024), highlighting its potential impact on evaluation fairness. Following the time series forecasting benchmark TFB's settings (Qiu et al. 2024), we do not use the "drop last" trick during the testing phase to ensure a fair comparison. Our benchmark comprehensively includes 20+ state-of-the-art forecasting models across 17 real-world datasets, ensuring a fair comparison by unifying implementation settings and avoiding over-tuning parameter. And some approaches (Hu et al. 2025b) discard the use of EarlyStopping, which in our view disregards the essential role of the validation set. Therefore, when integrating with other methods, we consistently retain this setting to ensure fair and reliable evaluation.

Unified Experimental Settings. To ensure fair comparison, we conduct two types of experiments. For long-term forecasting, we follow the unified evaluation protocol proposed by TimesNet (Wu et al. 2023), using a lookback length

Model Variant	**MSE**	**MAE**
Baseline	0.277	0.326
+ Adaptive Patch Embedding (APE)	0.258	0.302
+ APE and Segment-wise Prompt Tuning	**0.254**	**0.301**

Table 2: Ablation study of each component on five datasets. See datils in Appendix Table 14.

Granularity	**MSE**	**MAE**
[8, 16]	0.231	0.288
[8, 32]	**0.228**	**0.286**
[8, 16, 32]	0.232	0.290
[8, 16, 64]	0.230	0.289
[8, 32, 64]	0.235	0.289
[4, 8, 16, 32]	0.232	0.290
[8, 16, 32, 64]	**0.228**	**0.286**

Table 3: Impact of patch granularity on six datasets, see details in Appendix Table 13.

$L = 96$ and prediction lengths $T = 96, 192, 336, 720$ across all long-term datasets. For short-term forecasting, we adopt a fixed prediction length $T = 12$, which is commonly used in traffic prediction tasks. The lookback length is also set to $L = 96$. The averaged results under this unified setting are reported in Table 1. Table 1 presents results across all models using best configurations obtained via hyperparameter tuning, including the number of layers, attention heads, hidden dimension (d_{model}), and feedforward dimension (d_{ff}), with a lookback window of 96. In contrast, Table 5 reports results with an identical parameter setting shared by all models, using a longer lookback window of 320.

Ablation Experiments. We evaluate the contributions of each component in Table 2. Adaptive Patch Embedding (APE) improves performance by enabling input-aware tokenization, while segment-wise prompt tuning further enhances accuracy through localized decoding. Table 3 shows the results of different patch size combination. Figure 4

Methods		TimeMosaic		TimeMoE_l		TimeMoE_b		MOIRAI_l		MOIRAI_b		Chronos_b		Chronos_s		TimesFM		Moment	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.367	0.395	**0.350**	0.382	0.357	0.381	0.381	0.388	0.376	0.392	0.384	**0.379**	0.394	0.381	0.414	0.404	0.688	0.557
	192	0.395	0.412	0.388	0.412	**0.384**	**0.404**	0.434	0.415	0.412	0.413	0.441	0.412	0.455	0.414	0.465	0.434	0.688	0.560
	336	**0.410**	**0.423**	0.411	0.430	0.411	0.434	0.495	0.445	0.433	0.428	0.475	0.430	0.499	0.444	0.503	0.456	0.675	0.563
	720	**0.422**	**0.443**	0.427	0.455	0.449	0.477	0.611	0.510	0.447	0.444	0.472	0.446	0.520	0.476	0.511	0.481	0.683	0.585
	Avg	0.399	0.418	**0.394**	0.419	0.400	0.424	0.480	0.439	0.417	0.419	0.443	**0.416**	0.467	0.428	0.473	0.443	0.683	0.566
ETTm2	96	0.190	0.276	0.197	0.286	0.201	0.291	0.211	0.274	0.205	0.273	**0.177**	**0.244**	0.180	0.251	0.202	0.270	0.260	0.335
	192	**0.249**	0.313	0.250	0.322	0.258	0.334	0.281	0.318	0.275	0.316	0.251	**0.293**	0.251	0.298	0.289	0.321	0.289	0.350
	336	**0.305**	0.347	0.337	0.375	0.324	0.373	0.341	0.355	0.329	0.350	**0.305**	**0.327**	0.315	0.338	0.360	0.366	0.324	0.369
	720	0.396	0.407	0.480	0.461	0.488	0.464	0.428	0.428	0.437	0.411	0.419	**0.394**	0.421	**0.403**	0.462	0.430	**0.394**	0.409
	Avg	**0.285**	0.336	0.316	0.361	0.317	0.365	0.315	0.343	0.311	0.337	0.288	**0.314**	0.291	0.330	0.328	0.346	0.316	0.365

Table 4: Zero-shot forecasting results on two datasets. TimeMosaic is trained with an input length of 512 and an output horizon of 720. The symbols s, b, and l represent the small, base, and large versions, respectively. See details in Appendix Table 24.

Models	**TimeMosaic (Ours)**		SimpleTM (2025)		TimeFilter (2025)		xPatch (2025)		PatchMLP (2025)		DUET (2025)		iTransformer (2024)		TimeMixer (2024)		PatchTST (2023)		DLinear (2023)		FreTS (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.360	**0.377**	**0.350**	0.379	0.399	0.408	0.354	0.386	0.374	0.396	0.360	0.380	0.386	0.406	0.356	0.385	0.372	0.399	0.357	0380	0.369	0.391
ETTm2	**0.256**	**0.310**	0.263	0.318	0.278	0.330	0.262	0.322	0.269	0.324	0.257	0.314	0.293	0.338	0.265	0.323	0.278	0.332	0.282	0.348	0.284	0.339
ETTh1	**0.412**	**0.423**	0.428	0.438	0.511	0.482	0.446	0.445	0.462	0.460	0.414	0.429	0.487	0.475	0.430	0.438	0.503	0.485	0.429	0.444	0.461	0.465
ETTh2	0.363	0.393	0.377	0.404	0.402	0.421	0.354	0.397	0.380	0.416	**0.344**	**0.389**	0.411	0.425	0.358	0.396	0.398	0.427	0.492	0.478	0.495	0.483
Weather	0.231	**0.256**	0.230	0.267	0.231	0.270	**0.224**	0.262	0.230	0.267	0.248	0.280	0.237	0.272	0.227	0.266	0.236	0.271	0.246	0.299	0.230	0.277
Traffic	0.433	0.287	0.468	0.336	**0.420**	**0.284**	0.430	0.300	0.525	0.382	0.462	0.330	0.463	0.335	0.444	0.316	0.428	0.297	0.445	0.308	0.461	0.313
Electricity	0.170	0.263	0.180	0.277	**0.167**	**0.260**	0.170	0.264	0.200	0.302	0.185	0.287	0.180	0.277	0.173	0.267	0.173	0.272	0.170	0.269	0.171	0.270
Wind	0.774	**0.680**	0.779	0.690	0.815	0.703	0.762	0.682	0.772	0.689	0.762	0.683	0.807	0.701	0.762	0.684	0.811	0.710	0.744	0.681	**0.742**	**0.680**
Solar	0.226	**0.241**	0.272	0.314	**0.200**	0.255	**0.200**	0.253	0.250	0.288	0.265	0.289	0.232	0.280	0.224	0.276	0.221	0.262	0.255	0.315	0.212	0.267

Table 5: Averaged forecasting results under a unified and more fair evaluation setting with a farther lookback window. Compared to Table 1, Table 5 offers a fairer comparison by adopting an extended lookback window, which better supports models like DUET and TimeFilter to realize their potential. Moreover, we fix key hyperparameters such as `d_model`, `d_ff`, learning rate, and training epochs, avoiding test-set-based tuning and ensuring a fair evaluation. See details in Appendix G and Table 21.

demonstrates that segmented prediction consistently outperforms non-segmented baselines, while showing robustness across different segment sizes, see details in Appendix Figure 12. We adopt a fixed segment length of $L/3$ without cherry-picking.

Main Results. As shown in Table 1, TimeMosaic consistently achieves the best or second-best performance across all benchmarks. In the more fair comparison shown in Table 5, our method also achieves competitive performance. Furthermore, TimeMosaic demonstrates robust performance under longer lookback windows, as illustrated in Figure 6. See short-term forecasting results in Appendix Table 17. Table 6 compares the best hyperparameter configurations of seven SOTA models, where all models are tuned under the same search protocol. Due to the extremely high computational cost, each model requires exploring **10,800** different parameter settings, we only report results on seven models across five datasets.

Zero−shot. To assess cross-dataset generalization, we conduct zero-shot forecasting where models are trained on one dataset and tested on another. As shown in Appendix Table 23, TimeMosaic consistently achieves the best performance across all transfer settings, outperforming GPT4TS (Zhou et al. 2023), DLinear, and PatchTST. In addition, we further scale up TimeMosaic on the BLAST dataset (Shao et al. 2025) with an input length of 512 and an output horizon of 720. This large-scale setting (27M parameters, trained on two V100 GPUs for about 40 hours) allows a direct comparison with recent time series foundation models (TSFMs). The results (Appendix J, Tables 4 and 16) show that TimeMosaic achieves zero-shot performance on par with representative TSFMs such as TimeMoe (Shi et al. 2025), Moirai (Woo et al. 2024), Chronos (Ansari et al. 2024), TimesFM (Das et al. 2024), and Moment (Goswami et al. 2024) while maintaining a moderate model size and inference cost. This confirms that the architectural choices of adaptive patch embedding and segment-wise decoding remain effective even at the foundation-model scale.

Efficiency Analysis. Appendix Table 15 reports forecasting error and efficiency under the unified search protocol described in Appendix Section I. For each dataset and model, we first conduct hyperparameter search and select the configuration that achieves the lowest validation error averaged over MSE and MAE. We then measure parameter count,

Models	**TimeMosaic** Ours		SimpleTM (2025)		TimeFilter (2025b)		DUET (2025b)		iTransformer (2024)		TimeMixer (2024)		PatchTST (2023)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.342**	**0.367**	0.346	0.378	0.353	0.377	0.354	0.377	0.356	0.385	0.349	0.392	0.348	0.380	0.356	0.378
ETTm2	**0.250**	**0.305**	0.257	0.316	0.253	0.317	0.253	0.313	0.267	0.325	0.256	0.315	0.256	0.318	0.262	0.326
ETTh1	**0.397**	**0.417**	0.416	0.434	0.414	0.429	0.406	0.425	0.425	0.438	0.416	0.428	0.410	0.430	0.419	0.437
ETTh2	0.348	**0.383**	0.358	0.396	0.347	0.394	**0.334**	0.384	0.362	0.398	0.349	0.392	0.359	0.399	0.412	0.433
Weather	0.223	**0.251**	**0.220**	0.259	**0.220**	0.261	0.240	0.277	0.230	0.269	0.229	0.265	0.223	0.260	0.240	0.291

Table 6: Results under the hyperparameter search setting described in Appendix Section I. The lookback window is selected from $\{96, 192, 320, 512\}$, and the best configuration is reported for each model. This setup ensures that the comparison reflects each model's optimal performance rather than a fixed setting constraint. As far as we know, we are the first open source parameter search script. See details in Appendix Table 22.

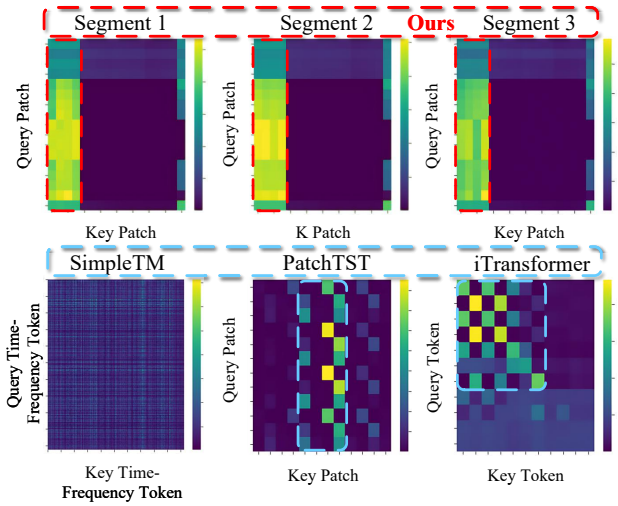

Figure 5: Attention Pattern Visualization Across Models.

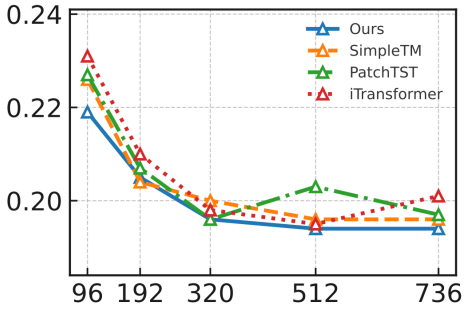

Figure 6: Sensitivity of multivariate forecasting performance to look-back window size (MSE). We evaluate four models across five input lengths ($T = 192$) on Weather. See more details in Appendix G.

inference latency, and GPU memory usage at this setting. This protocol anchors every method at its best attainable error level within the same search space and avoids biased comparisons that could arise from deliberately changing hyperparameters to influence efficiency. The results show that TimeMosaic achieves competitive or leading error on all five datasets, for example attaining the lowest MSE on ETTm2 and the lowest MAE on Weather, while keeping parameter scale in the range of a few hundred thousand, which is substantially smaller than SimpleTM or iTransformer. Although its inference time is higher than that of extremely lightweight models such as iTransformer, it remains within a reasonable margin and consistently uses moderate memory. The slightly slower inference mainly arises from the segment-wise step-by-step decoding design, and this behavior is consistent with our experimental observations. Overall, TimeMosaic presents a balanced error–efficiency performance, showing that its gains stem from architectural design rather than hyperparameter tuning.

Visualization. Existing models differ in their attention interaction schemes. PatchTST slices each variable into temporal patches and applies self-attention across patches within the same variable. iTransformer treats each variable as a token and performs self-attention across channels. In contrast, SimpleTM constructs time-frequency tokens via wavelet transforms and enables joint attention across both time and channels. Our TimeMosaic instead interacts primarily through learnable *prompt embeddings* assigned to different forecast segments, guiding the model to specialize decoding over distinct temporal intervals in Figure 5.

5 Conclusion

We proposed TimeMosaic, a forecasting framework that jointly addresses the dual challenges of encoding heterogeneity and decoding heterogeneity in multivariate time series. Through adaptive patch embedding, TimeMosaic dynamically adjusts granularity according to local information density, ensuring both motif reuse and structural clarity while maintaining temporal coherence. Complementarily, segment-wise prompt tuning enables horizon-aware forecasting by treating different prediction segments as related subtasks within a unified multi-task paradigm. Extensive experiments verify that TimeMosaic achieves consistent improvements in forecasting accuracy and efficiency. In future work, we will extend TimeMosaic to streaming scenarios and investigate its potential for large-scale pretraining towards time series foundation modeling.

Acknowledgments

This research was partly supported by the National Natural Science Foundation of China under Grant Nos. U24A20232 and 62402475.

References

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Pineda Arango, S.; Kapoor, S.; Zschiegner, J.; Maddix, D. C.; Mahoney, M. W.; Torkkola, K.; Gordon Wilson, A.; Bohlke-Schneider, M.; and Wang, Y. 2024. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research (TMLR)*.

Axtell, R. L. 2001. Zipf Distribution of U.S. Firm Sizes. *Science*, 293(5536): 1818–1820.

Cao, J.; Li, J.; Yang, Z.; and Zhou, R. 2024. Enhanced Multimodal Aspect-Based Sentiment Analysis by LLM-Generated Rationales. In *International Conference on Neural Information Processing*, 228–243. Springer.

Chen, H.; Luong, V.; Mukherjee, L.; and Singh, V. 2025. SimpleTM: A Simple Baseline for Multivariate Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*.

Chen, P.; Zhang, Y.; Cheng, Y.; Shu, Y.; Wang, Y.; Wen, Q.; Yang, B.; and Guo, C. 2024. Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

Crawshaw, M. 2020a. Multi-Task Learning with Deep Neural Networks: A Survey.

Crawshaw, M. 2020b. Multi-Task Learning with Deep Neural Networks: A Survey. *CoRR*, abs/2009.09796.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Ding, K.; Fan, F.; Wang, Y.; Jian, R.; Wang, X.; Gong, L.; Jiang, Y.; Luo, C.; and Zhan, J. 2025. DualSG: A Dual-Stream Explicit Semantic-Guided Multivariate Time Series Forecasting Framework. In *Proceedings of the 33rd ACM International Conference on Multimedia*.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. MOMENT: A Family of Open Time-series Foundation Models. In *International Conference on Machine Learning (ICML)*.

Hu, Y.; Li, Y.; Liu, P.; Zhu, Y.; Li, N.; Dai, T.; tao Xia, S.; Cheng, D.; and Jiang, C. 2025a. FinTSB: A Comprehensive and Practical Benchmark for Financial Time Series Forecasting. *arXiv preprint arXiv:2502.18834*.

Hu, Y.; Zhang, G.; Liu, P.; Lan, D.; Li, N.; Cheng, D.; Dai, T.; Xia, S.-T.; and Pan, S. 2025b. TimeFilter: Patch-Specific Spatial-Temporal Graph Filtration for Time Series Forecasting. In *International Conference on Machine Learning*.

Huang, M.; Mao, Z.; Chen, Z.; and Zhang, Y. 2023. Towards Accurate Image Coding: Improved Autoregressive Image Generation With Dynamic Vector Quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 22596–22605.

Huang, Q.; Zhou, Z.; Yang, K.; Yi, Z.; Wang, X.; Jiang, W.; and Wang, Y. 2025. TimeBase: The Power of Minimalism in Long-term Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*.

Jin, M.; Wang, S.; Ma, L.; Chu, Z.; Zhang, J. Y.; Shi, X.; Chen, P.-Y.; Liang, Y.; Li, Y.-F.; Pan, S.; and Wen, Q. 2024. Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Kudrat, D.; Xie, Z.; Sun, Y.; Jia, T.; and Hu, Q. 2025. Patch-wise Structural Loss for Time Series Forecasting. In *ICML*.

Laban, P.; Kryscinski, W.; Agarwal, D.; Fabbri, A.; Xiong, C.; Joty, S.; and Wu, C.-S. 2023. SummEdits: Measuring LLM Ability at Factual Reasoning Through The Lens of Summarization. In *EMNLP*.

Lai, G.; Chang, W.-C.; Yang, Y.; and Liu, H. 2018. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *ACM SIGIR*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021. GPT Understands, Too. *arXiv:2103.10385*.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2024. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

Liu, Y.; Qin, G.; Shi, Z.; Chen, Z.; Yang, C.; Huang, X.; Wang, J.; and Long, M. 2025. Sundial: A Family of Highly Capable Time Series Foundation Models. In *International Conference on Machine Learning*.

Nate Gruver, S. Q., Marc Finzi; and Wilson, A. G. 2023. Large Language Models Are Zero Shot Time Series Forecasters. In *Advances in Neural Information Processing Systems*.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; and Yang, B. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proc. VLDB Endow.*, 17(9): 2363–2377.

Qiu, X.; Wu, X.; Cheng, H.; Liu, X.; Guo, C.; Hu, J.; and Yang, B. 2025a. DBLoss: Decomposition-based Loss Function for Time Series Forecasting. In *NeurIPS*.

Qiu, X.; Wu, X.; Lin, Y.; Guo, C.; Hu, J.; and Yang, B. 2025b. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*, 1185–1196.

Qiu, X.; Zhu, Y.; Li, Z.; Cheng, H.; Wu, X.; Guo, C.; Yang, B.; and Hu, J. 2025c. DAG: A dual causal network for time series forecasting with exogenous variables. *arXiv preprint arXiv:2509.14933*.

Rousseeuw, P. J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20: 53–65.

Ruder, S. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. arXiv:1706.05098.

Shao, Z.; Li, Y.; Wang, F.; Yu, C.; Fu, Y.; Qian, T.; Xu, B.; Diao, B.; Xu, Y.; and Cheng, X. 2025. BLAST: Balanced Sampling Time Series Corpus for Universal Forecasting Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*.

Shi, X.; Wang, S.; Nie, Y.; Li, D.; Ye, Z.; Wen, Q.; and Jin, M. 2025. Time-MoE: Billion-Scale Time Series Foundation Models with Mixture of Experts. In *International Conference on Learning Representations (ICLR)*.

Stitsyuk, A.; and Choi, J. 2025. xPatch: Dual-Stream Time Series Forecasting with Exponential Seasonal-Trend Decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tang, P.; and Zhang, W. 2025. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *AAAI*.

Wang, H.; Pan, L.; Chen, Z.; Yang, D.; Zhang, S.; Yang, Y.; Liu, X.; Li, H.; and Tao, D. 2025a. FreDF: Learning to Forecast in the Frequency Domain. In *ICLR*.

Wang, S.; Li, J.; Shi, X.; Ye, Z.; Mo, B.; Lin, W.; Ju, S.; Chu, Z.; and Jin, M. 2025b. TimeMixer++: A General Time Series Pattern Machine for Universal Predictive Analysis. In *International Conference on Learning Representations (ICLR)*.

Wang, S.; Wu, H.; Shi, X.; Hu, T.; Luo, H.; Ma, L.; Zhang, J. Y.; and ZHOU, J. 2024. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*.

Warren Liao, T. 2005. Clustering of time series data—a survey. *Pattern Recognition*, 38(11): 1857–1874.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *Forty-first International Conference on Machine Learning (ICML)*.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Xia, Y.; Fu, F.; Zhang, W.; Jiang, J.; and Cui, B. 2024. Efficient Multi-task LLM Quantization and Serving for Multiple LoRA Adapters. In *Advances in Neural Information Processing Systems*.

Xie, Y.; Xiong, Y.; Shi, Z.; Niu, H.; and Liu, Z. 2025. The language of time: a language model perspective on time-series foundation models. arXiv:2507.00078.

Yi, K.; Zhang, Q.; Fan, W.; Wang, S.; Wang, P.; He, H.; An, N.; Lian, D.; Cao, L.; and Niu, Z. 2023. Frequency-domain MLPs are More Effective Learners in Time Series Forecasting. In *Advances in Neural Information Processing Systems*.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, T.; Zhang, Y.; Cao, W.; Bian, J.; Yi, X.; Zheng, S.; and Li, J. 2022. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. arXiv:2207.01186.

Zhang, W.; Deng, Y.; Liu, B.; Pan, S.; and Bing, L. 2024. Sentiment Analysis in the Era of Large Language Models: A Reality Check. In *NAACL*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhou, T.; Niu, P.; Wang, X.; Sun, L.; and Jin, R. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Advances in Neural Information Processing Systems*.