DualMixFormer: A Dual-Mixing Transformer for Long-Term Multivariate Time Series Forecasting

1st Ruiqi Liu
School of Artificial Intelligence
South China Normal University
Guangzhou, China
aatroxemail@qq.com

2nd Dingju Zhu*
School of Artificial Intelligence
South China Normal University
Guangzhou, China
zhudingju@m.scnu.cn

***Abstract*—Long-term multivariate time series forecasting (LTSF) demands accurate long-horizon prediction while modeling cross-variable dependencies. Most Transformer-based methods focus attention along the temporal axis, underutilizing inter-variable interactions and suffering from error accumulation over long horizons. We propose DualMixFormer, a variable-centric framework that treats each variable as a token and applies self-attention across variables. To enhance token representations amid heterogeneous periodicity, we introduce MEVA, a multi-embedding module that injects channel identity and phase embeddings to guide cross-variable attention explicitly. For robustness, DualMixFormer employs a dual-branch head: a lightweight MLP branch provides a low-variance continuation bias for smooth components, while a MEVA-conditioned Transformer branch models nonlinear residuals. Predictions are fused via learnable channel-wise weights, adaptively combining smooth and residual components. RevIN is integrated to address instance-wise distribution shifts. Experiments on seven benchmarks show competitive performance across horizons. Ablations and visualizations confirm the model's effectiveness, interpretability, and robustness.**

***Index Terms*—long-term time series forecasting, multivariate cross-variable modeling, dual-branch hybrid architecture**

I. Introduction

Long-term time series forecasting (LTSF) estimates future observations over extended horizons and supports operational decision systems such as electricity load scheduling, traffic regulation, meteorological monitoring, and financial risk assessment [1]. Compared with short-horizon prediction, LTSF is more challenging because errors can compound with horizon length, long-range dependencies must be captured under limited context, and instance-wise distribution shifts often violate stationarity assumptions. In multivariate settings, cross-variable information further varies across datasets: engineered systems (e.g., power grids or transportation networks) may exhibit strong coupling, whereas other domains show sparse or time-varying dependencies. Thus, effective LTSF models should exploit inter-variable interactions when informative while maintaining stable long-horizon extrapolation.

Recent progress has been driven by deep learning, including CNN-based forecasters, linear methods, and Transformer variants [2]–[7]. Transformer-based models are often favored because self-attention enables flexible information aggregation [7]–[10]; however, many early approaches apply attention mainly along the temporal axis (*cross-time* attention), which is costly under long look-back windows and whose necessity for trend extrapolation has been questioned, e.g., by the mask-series analysis in DLinear [11]. With *variate tokenization*, an alternative view emerges: variables are treated as tokens and attention is redirected to *cross-variable* dependencies. This direction is supported by iTransformer [12], showing cross-variable attention can provide a strong backbone for multivariate forecasting.

Under variate tokenization, two gaps become salient for long-horizon prediction. First, compressing a historical segment into a variable token makes auxiliary conditioning essential: the model should know *which* variable a token represents, and can benefit from lightweight periodic-stage cues because explicit temporal-stage information is no longer token-level. Second, long horizons exacerbate the bias–variance tension. High-capacity nonlinear models can fit complex interactions but may extrapolate unstably and amplify errors; a lightweight MLP pathway can provide lower-variance continuation for smooth components with modest flexibility beyond strict linearity, but may be insufficient when predictive signal lies in nonlinear cross-variable residuals. Hence, cross-variable modeling and stable extrapolation should be addressed jointly.

To this end, we propose **DualMixFormer**, a hybrid framework combining a variate-tokenized cross-variable Transformer with conditioning and robustness mechanisms. **MEVA** (Multi-Embedding Variable Attention) augments inverted variable tokens with **channel embeddings** for variable identity and **phase embeddings** for periodic-stage information derived from the last observed step, reducing representation ambiguity and enabling the Transformer branch to focus on *nonlinear cross-variable residuals*. For forecasting, a dual-branch head is used: the MEVA-conditioned Transformer provides residual correction, while an integrated *lightweight MLP branch* supplies a low-variance continuation bias for smooth trends and other regular components. Their outputs are fused via learnable channel-wise weights, realizing the principle that *smooth low-frequency dynamics are backed by the lightweight MLP pathway, while residual deviations are corrected by attention*. RevIN [13] is further applied to mitigate instance-wise distri-

*Corresponding author:Dingju Zhu (zhudingju@m.scnu.cn)

bution shift through reversible normalization. DualMixFormer is evaluated on seven multivariate benchmarks across multiple horizons, showing competitive performance. Ablations and analyses (attention visualization, correlation characterization, and random masking) further verify the contribution of each component.

In summary, the contribution of this study is threefold:

- A variate-tokenization embedding augmentation module, **MEVA**, is proposed. By injecting **channel** identity and **phase** context, MEVA strengthens variable-token representations and improves the selectivity of cross-variable attention.
- A **dual-branch forecasting head** is designed, in which a MEVA-conditioned cross-variable Transformer branch is used for nonlinear cross-variable residual modeling, while a lightweight **MLP extrapolation branch** provides a low-variance continuation bias for smooth components; their outputs are fused by learnable channel-wise weights to enhance long-horizon robustness.
- Comprehensive **experiments and analyses** are conducted, including ablations, attention map visualization, correlation-structure analysis, and robustness evaluation under random masking, to verify effectiveness and interpret performance behavior.

Code availability. The code is currently not publicly available and will be released in the future.

II. Related Work

A. Non-Transformer models

Classical statistical methods, including ARMA and ARIMA [1], have long been used for time series forecasting. Their effectiveness largely relies on the assumption that future observations can be approximated by (approximately) linear relations of past values, making them suitable for smooth trend-like dynamics in relatively stationary regimes. With the availability of larger datasets and advances in representation learning, research has increasingly shifted toward deep learning, since neural architectures can learn task-specific features and capture interaction patterns that are difficult to encode manually [14]. Representative families include multilayer perceptrons (MLPs) [15], recurrent neural networks (RNNs) [2], and convolutional neural networks (CNNs) [16], as well as distribution-aware designs such as heteroscedastic neural networks (HNNs) [17] and quantile-based ensembles [18].

Different architectures encode distinct inductive biases. DeepAR combines recurrent dynamics with autoregressive likelihood modeling to capture probabilistic characteristics [4], [19]. Temporal convolutional networks (TCNs) use causal convolutions to encode temporal causality and long receptive fields [3], [5], [20]–[22]. LSTNet integrates CNN and RNN components to extract short-term structure and longer-range correlations [23], while MTGNN incorporates graph convolution to model cross-variable relations jointly with temporal dependencies [24]. Attention-augmented recurrent designs such as TPA-LSTM add attention on top of LSTM backbones to improve performance [25]. In the LTSF setting, *simple low-variance predictors* have regained attention: DLinear shows that carefully constructed linear mappings can be competitive under long horizons [11], highlighting the value of stable extrapolation for trend and regular components. In the same spirit, a *lightweight MLP branch* (e.g., a compact two-layer MLP) can provide a robust inductive bias while retaining more flexibility than strict linearity. LightTS similarly emphasizes efficient extraction of informative temporal structure via sampling [6]. TimesNet converts 1D sequences into period-aware 2D representations and applies CNN-based extraction, achieving strong performance across LTSF and related tasks [26].

B. Transformer-based forecasting

Transformer [7] has achieved broad success across natural language processing, speech, and vision [27]–[30]. Accordingly, many Transformer variants have been proposed for time series forecasting, often aiming to improve attention efficiency while preserving capacity. LogTrans uses a LogSparse pattern to reduce memory usage while maintaining local sensitivity [31]. Informer [8] and Autoformer [9] reduce complexity (e.g., to $O(L \log L)$), while FEDformer [10] performs frequency-domain modeling and achieves linear-time behavior by selecting frequency bases. To address non-stationarity, Non-stationary Transformer introduces stationarization modules and De-stationary Attention [32]. PatchTST partitions sequences into patches and applies attention across segments [33].

Despite these advances, the dominant paradigm remains *cross-time* attention, allocating modeling capacity mainly along the temporal dimension. Its necessity for long-horizon trend learning has been questioned; DLinear reports mask-series evidence suggesting cross-time attention may be unnecessary for trend extrapolation [11]. Crossformer proposes a Two-Stage Attention module to jointly model cross-time and cross-dimension dependencies [34]. However, its performance is often comparable to or weaker than strong linear baselines, indicating that attention alone does not fully resolve long-horizon stability. This motivates hybrid designs that reserve a *low-variance lightweight MLP branch* for smooth components, while using attention mainly for cross-variable residual interactions.

C. Embedding techniques under variate tokenization

Embeddings are widely used in Transformer models to encode semantic distinctions without changing the backbone, as in sentence-level representations in BERT and position-related embeddings in ViT [27], [35]. In multivariate forecasting, token embeddings are typically implemented as projections of raw observations [12], [34]. Auxiliary embeddings have been studied more under temporal tokenization; for example, Informer uses timestamp embeddings for calendrical cues, and positional embeddings represent temporal ordering [8], [9].

With the increasing adoption of variate tokenization, embeddings shift from encoding temporal order to conditioning channel-level tokens. Recent work has explored inductive

biases under this setting. CycleNet introduces cycle-related parameters for residual cycle forecasting in an MLP-based framework [36], though gains on Transformer backbones are not always consistent. Recent variate-tokenization Transformers (e.g., iTransformer) typically project each channel sequence into a token representation while leaving additional conditioning signals underexplored [12]. This motivates lightweight auxiliary embeddings that supply channel identity and periodic-stage cues without changing the core architecture. Such conditioning is especially relevant for hybrid heads: once trend continuation is stabilized by a low-variance *lightweight MLP branch*, the Transformer branch can focus on nonlinear cross-variable residuals, supporting a "trend-by-MLP, residual-by-attention" decomposition under long horizons.

III. Methodology

Long-term time series forecasting (LTSF) aims to predict the future values of a multivariate time series over an extended horizon. Given a historical series

$$\mathcal{H} = \{X_t^1, \ldots, X_t^N\}_{t=1}^{L},$$

where L denotes the look-back window size and N is the number of variables, with X_t^i representing the value of the i-th variable at time step t, the task is to forecast the future series

$$\mathcal{F} = \{X_t^1, \ldots, X_t^N\}_{t=L+1}^{L+T},$$

where T denotes the forecasting horizon.

In this section, we first introduce the overall architecture of **DualMixFormer**. Subsequently, we delineate its key components: the **Multi-Embedding Variable Attention (MEVA)** module (inverted embedding with channel and phase embeddings), the **cross-variable Transformer encoder**, the **integrated lightweight MLP branch** with a mixed-output head, and the **RevIN** normalization module. The core idea of DualMixFormer is to replace cross-time attention with *cross-variable* attention, while conditioning variable-wise representations with channel identity and phase context, and integrating a lightweight MLP branch to provide a low-variance extrapolation bias for trend-like and regular components.

A. Overall DualMixFormer Architecture

DualMixFormer consists of two parallel branches: a *lightweight MLP branch* that provides a low-variance continuation for trend-like and regular components, and a Transformer branch that models nonlinear cross-variable dependencies. As shown in Fig. 1, the input is first normalized by RevIN, then fed into the MEVA-based cross-variable Transformer encoder and the lightweight MLP branch. Their predictions are fused via learnable channel-wise weights to produce the final forecast, combining the robustness of smooth extrapolation with attention-based dependency modeling.

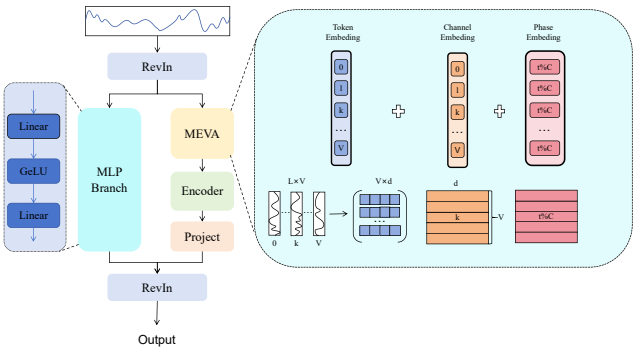


Fig. 1. Overall architecture of **DualMixFormer**. The model combines a MEVA-based cross-variable Transformer branch for inter-variable dependency modeling with a lightweight channel-independent MLP branch for low-frequency components, and fuses their outputs via learnable channel-wise weights.

B. Cross-Variable Transformer in DualMixFormer

The Transformer branch performs *cross-variable* attention over variate tokens. Given $X \in \mathbb{R}^{B\times L\times N}$, MEVA converts each variable trajectory into a d-dimensional token and adds channel identity and phase context:

$$H^{(0)} = \mathrm{LN}\big(\mathrm{Embed}(X) + E_c + E_p\big) \in \mathbb{R}^{B\times N\times d}, \quad (1)$$

where $\mathrm{LN}(\cdot)$ stabilizes optimization after embedding fusion.

a) Channel embedding.: A learnable table $\alpha_c \in \mathbb{R}^{N\times d}$ encodes channel identity. The i-th channel embedding is

$$E_c^{(i)} = \mathrm{Lookup}(\alpha_c, i) \in \mathbb{R}^{1\times d}, \quad (2)$$

and stacking $\{E_c^{(i)}\}_{i=1}^{N}$ yields $E_c \in \mathbb{R}^{N\times d}$, broadcast over the batch.

b) Phase embedding.: Given period length P, a learnable table $\alpha_p \in \mathbb{R}^{P\times d}$ is used. Let t be the absolute index of the last observed step; the phase embedding is

$$E_p = \mathrm{Lookup}(\alpha_p,\, t \bmod P) \in \mathbb{R}^{1\times d}, \quad (3)$$

shared within each sample and broadcast over variables.

c) Encoder blocks.: Each encoder layer applies MHA and FFN to $H^{(\ell)} \in \mathbb{R}^{B\times N\times d}$:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad (4)$$

with $Q = H^{(\ell)}W_Q$, $K = H^{(\ell)}W_K$, and $V = H^{(\ell)}W_V$, producing explicit inter-variable interaction weights over the N tokens.

d) Residual projection head (encoder output → horizon).: After M layers,

$$H^{\mathrm{enc}} = \mathrm{Encoder}\big(H^{(0)}\big) \in \mathbb{R}^{B\times N\times d}. \quad (5)$$

A residual projection head maps features to the horizon:

$$Y_{\mathrm{nl}} = \mathrm{Proj}\big(H^{\mathrm{enc}} + H^{(0)}\big) \in \mathbb{R}^{B\times N\times T}. \quad (6)$$

We instantiate $\mathrm{Proj}(\cdot)$ as an MLP:

$$\mathrm{Proj}(\cdot) = W_3\,\sigma\Big(\mathrm{Drop}\big(W_2\,\sigma(\mathrm{Drop}(W_1(\cdot)))\big)\Big), \quad (7)$$

TABLE I
STATISTICS OF THE BENCHMARK TIME-SERIES DATASETS

Dataset	Channels	Time steps	Frequency	Period
ETTh1	7	17,420	1 hour	24
ETTh2	7	17,420	1 hour	24
ETTm1	7	69,680	15 mins	96
ETTm2	7	69,680	15 mins	96
ECL	321	26,304	1 hour	168
Traffic	862	17,544	1 hour	168
Weather	21	52,696	10 mins	144

where $\sigma(\cdot)$ is GELU and $\mathrm{Drop}(\cdot)$ is dropout, with widths $d \to 2d \to 4d \to T$.

e) Permutation to standard output format.:

$$\widehat{Y}_{\mathrm{nl}} = \mathrm{Permute}\big(Y_{\mathrm{nl}}\big) \in \mathbb{R}^{B\times T\times N}. \tag{8}$$

C. Lightweight MLP Integration and RevIN Modules

a) Lightweight MLP branch.: The MLP branch is channel-independent and parameter-efficient. We permute $X \in \mathbb{R}^{B\times L\times N}$ to $X^\top \in \mathbb{R}^{B\times N\times L}$ and apply a shared two-layer MLP to each channel:

$$Y_{\mathrm{mlp}} = \mathbf{W}_2\,\mathrm{GELU}\big(\mathbf{W}_1 X^\top + \mathbf{b}_1\big) + \mathbf{b}_2 \in \mathbb{R}^{B\times N\times T}, \tag{9}$$

where $\mathbf{W}_1 \in \mathbb{R}^{r\times L}$, $\mathbf{W}_2 \in \mathbb{R}^{T\times r}$, and r is a small hidden width.

b) Branch integration.: Let $Y_{\mathrm{nl}} \in \mathbb{R}^{B\times N\times T}$ be the Transformer prediction. The mixed forecast is

$$Y_{\mathrm{raw}} = w_{\mathrm{tr}} \odot Y_{\mathrm{nl}} + w_{\mathrm{mlp}} \odot Y_{\mathrm{mlp}}, \tag{10}$$

where $w_{\mathrm{tr}}, w_{\mathrm{mlp}} \in \mathbb{R}^N$ are learnable channel-wise weights.

c) RevIN module.: DualMixFormer uses RevIN [13] to reduce instance-wise scale and shift variations:

$$\widehat{X} = \mathrm{RevIN}(X) = \gamma \cdot \frac{X - \mu(X)}{\sigma(X) + \epsilon} + \beta, \tag{11}$$

where $\mu(X)$ and $\sigma(X)$ are computed per sample and per channel. After mixing,

$$\widetilde{Y} = \mathrm{RevIN}^{-1}(Y_{\mathrm{raw}}), \tag{12}$$

yielding the final prediction $\widetilde{Y}$.

IV. EXPERIMENTS

A. Data and experimental setting

a) Dataset.: We evaluate on seven benchmarks: Electricity (ECL) [8], the ETT series [8], Traffic [8], and Weather [9]. We follow standard preprocessing settings (Liu et al. 2022, 2024c; Lin et al. 2024b, 2025), including train/validation/test splits and z-score normalization, and report MSE and MAE.

We study the period length P with daily and weekly candidates. While a weekly period can cover daily patterns, it yields fewer samples per phase and may hinder learning. We therefore choose P by empirical performance; period statistics are summarized in Table I.

b) Baselines: We compare DualMixFormer with recent state-of-the-art methods, including iTransformer [12], PatchTST [33], DLinear [11], Informer [8], and TimesNet [26].

c) Implementation details: Across all methods, optimization is performed using AdamW. The batch size is set to 32, and training is run for up to 10 epochs with early stopping (patience=3). A look-back window of 336 time steps is used throughout. All experiments are executed on a single NVIDIA GeForce RTX 3090 GPU (24GB).

B. Long-term Forecasting Results (DualMixFormer)

Table II compares the multivariate long-term forecasting performance of **DualMixFormer** with several representative baselines, including *PatchTST*, *DLinear*, *iTransformer*, *Informer*, and *TimesNet*, across seven real-world datasets under multiple prediction horizons, totaling 28 evaluation settings. DualMixFormer achieves robust and consistent improvements: it reduces MSE by an average of 1.98% and MAE by 3.96% over *PatchTST*, with gains observed in 21/28 (MSE) and 25/28 (MAE) settings. Against the simpler *DLinear*, the average reductions widen to 5.22% (MSE) and 7.69% (MAE), with wins in 21/28 and 24/28 cases, respectively. These results underscore that the proposed dual-branch architecture introduces transferable inductive biases that generalize across diverse temporal dynamics—not merely on a few favorable configurations. Notably, the largest gains occur on datasets exhibiting strong cross-variable coupling and structured residual patterns, while performance remains stable even when nonlinear signals are weak. On highly linear-dominated regimes like **ETTh1**, DualMixFormer matches *DLinear* closely, confirming that its residual modeling incurs minimal overhead when unnecessary. Even on challenging benchmarks such as **Traffic** and **ETTh2**, where squared-error metrics can be skewed by rare spikes, DualMixFormer maintains clear advantages in MAE—highlighting its effectiveness in reducing typical prediction errors.

C. Ablation Study

Table III evaluates key components of **DualMixFormer** by removing each one on **ETTm2**, **Electricity**, and **Traffic** under the same horizons. The resulting accuracy changes are consistent and component-specific, indicating that these modules provide complementary inductive biases for long-horizon forecasting.

a) Effect of the lightweight MLP branch: Removing the lightweight MLP branch degrades **ETTm2** and **Traffic**: averaged over settings, MSE increases by about 5.2% and 5.0%, and MAE on **ETTm2** rises by roughly 1.7%. This supports the MLP pathway as a low-variance continuation prior that curbs large-error accumulation (more visible in MSE). In contrast, **Electricity** slightly improves without the branch (about 1.0% lower MSE and 1.7% lower MAE), suggesting that a strong smooth-extrapolation bias can be suboptimal for fluctuation-dominated dynamics. On **Traffic**, MAE drops by about 4.6% despite higher MSE, consistent with reduced typical errors but less suppression of rare spikes.

TABLE II
MAIN RESULTS ON MULTIVARIATE LTSF BENCHMARKS. THE BEST RESULTS ARE IN BOLD AND THE SECOND BEST ARE UNDERLINED.

Dataset	L	DualMixFormer		PatchTST		DLinear		iTransformer		Informer		TimesNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.149**	**0.188**	0.158	0.206	0.176	0.237	0.159	0.209	0.228	0.311	0.163	0.218
	192	**0.193**	**0.231**	0.205	0.253	0.220	0.282	0.206	0.250	0.389	0.425	0.213	0.260
	336	**0.239**	**0.269**	0.259	0.294	0.265	0.319	0.253	0.289	0.583	0.543	0.280	0.306
	720	0.324	**0.327**	0.333	0.348	**0.323**	0.362	0.326	0.338	0.916	0.705	0.353	0.358
ETTh1	96	0.377	0.400	0.390	0.415	**0.375**	**0.399**	0.386	0.405	0.941	0.769	0.384	0.402
	192	0.422	0.423	0.431	0.438	**0.405**	**0.416**	0.441	0.436	1.007	0.786	0.436	0.429
	336	**0.435**	0.444	0.448	0.453	0.439	**0.443**	0.459	0.458	1.038	0.784	0.484	0.469
	720	0.471	**0.480**	**0.467**	0.485	0.472	0.490	0.500	0.491	1.144	0.857	0.521	0.500
ETTh2	96	0.287	**0.343**	**0.285**	0.349	0.289	0.353	0.297	0.349	1.549	0.952	0.340	0.374
	192	**0.351**	**0.384**	0.353	0.394	0.383	0.418	0.380	0.400	3.790	1.542	0.402	0.414
	336	0.364	0.404	**0.344**	**0.395**	0.448	0.465	0.417	0.432	2.769	1.297	0.399	0.431
	720	**0.391**	**0.433**	0.394	0.439	0.605	0.551	0.427	0.445	3.656	1.595	0.454	0.468
ETTm1	96	**0.292**	**0.335**	0.302	0.356	0.299	0.343	0.309	0.361	0.626	0.560	0.327	0.374
	192	0.341	**0.365**	0.345	0.384	**0.335**	**0.365**	0.344	0.382	0.725	0.619	0.374	0.387
	336	**0.368**	**0.383**	0.381	0.408	0.369	0.386	0.379	0.403	0.878	0.720	0.410	0.411
	720	0.426	**0.417**	0.437	0.441	**0.425**	0.421	0.438	0.436	1.133	0.845	0.458	0.450
ETTm2	96	**0.162**	**0.244**	0.173	0.265	0.167	0.260	0.178	0.264	0.355	0.462	0.187	0.267
	192	0.228	**0.290**	0.232	0.304	**0.224**	0.303	0.241	0.309	0.595	0.586	0.241	0.309
	336	**0.272**	**0.319**	0.289	0.342	0.281	0.342	0.306	0.348	0.990	0.766	0.304	0.349
	720	**0.362**	**0.376**	0.382	0.408	0.397	0.421	0.382	0.399	2.849	1.267	0.390	0.403
ECL	96	0.130	**0.222**	0.135	0.231	0.140	0.237	**0.132**	0.228	0.304	0.393	0.168	0.272
	192	**0.148**	**0.238**	0.154	0.250	0.153	0.249	0.158	0.251	0.327	0.417	0.184	0.289
	336	**0.167**	**0.256**	0.174	0.271	0.169	0.267	0.169	0.265	0.333	0.422	0.198	0.300
	720	0.205	**0.290**	0.210	0.303	**0.203**	0.301	0.207	0.300	0.333	0.413	0.219	0.316
Traffic	96	0.375	**0.250**	**0.367**	0.251	0.413	0.287	0.395	0.268	0.467	0.375	0.594	0.321
	192	0.402	0.262	**0.385**	**0.259**	0.424	0.290	0.417	0.276	0.455	0.371	0.604	0.324
	336	0.415	0.275	**0.398**	**0.265**	0.438	0.299	0.433	0.283	0.462	0.378	0.616	0.324
	720	0.441	**0.281**	**0.434**	0.287	0.466	0.374	0.467	0.302	0.495	0.400	0.671	0.349

TABLE III
ABLATION RESULTS ON ETTM2, ELECTRICITY, AND TRAFFIC (MSE/MAE). THE BEST RESULTS ARE IN BOLD.

Method	L	ETTm2		Electricity		Traffic	
		MSE	MAE	MSE	MAE	MSE	MAE
DualMixFormer	96	**0.162**	**0.244**	**0.125**	0.222	**0.381**	0.250
	192	**0.228**	**0.290**	**0.148**	**0.238**	**0.402**	0.262
	336	**0.272**	**0.319**	0.167	0.256	**0.415**	0.275
	720	**0.362**	**0.376**	0.205	0.290	**0.441**	0.281
w/o Lightweight MLP Branch	96	0.167	0.246	0.130	**0.220**	0.397	**0.238**
	192	0.233	**0.290**	0.150	**0.238**	0.418	**0.249**
	336	0.293	0.329	**0.166**	**0.255**	0.435	**0.256**
	720	0.390	0.387	**0.187**	**0.274**	0.472	**0.276**
w/o Embedding Augmentation	96	0.180	0.264	0.148	0.240	0.395	0.268
	192	0.250	0.309	0.162	0.253	0.417	0.276
	336	0.311	0.348	0.178	0.269	0.433	0.283
	720	0.412	0.407	0.225	0.317	0.467	0.302
w/o RevIN	96	0.187	0.275	0.154	0.250	0.411	0.279
	192	0.260	0.321	0.168	0.263	0.434	0.287
	336	0.323	0.362	0.185	0.280	0.450	0.294
	720	0.428	0.423	0.234	0.330	0.486	0.314

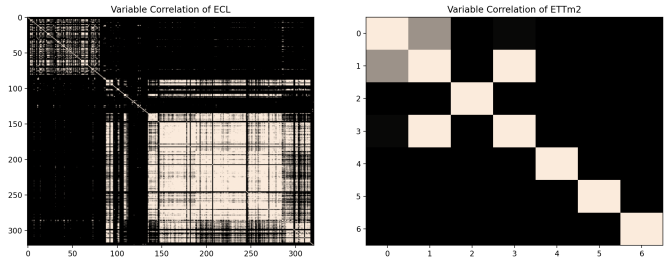
b) Effect of MEVA embedding augmentation: MEVA is uniformly beneficial. Removing it increases errors on all datasets: **ETTm2** (+12.2%/+8.0% MSE/MAE), **Electricity** (+11.1%/+7.2%), and **Traffic** (+4.4%/+5.7%), i.e., about +9.2% MSE and +7.0% MAE overall. This supports MEVA's role in providing explicit channel identity and phase context, reducing token ambiguity so cross-variable attention can focus on informative residual interactions.

c) Effect of RevIN: RevIN yields the strongest robustness gain. Without RevIN, MSE/MAE increase by about 16.6%/12.3% on **ETTm2**, 15.4%/11.6% on **Electricity**, and 8.6%/9.9% on **Traffic**, averaging to roughly +13.5% MSE and +11.3% MAE. This confirms RevIN as a key stabilizer under non-stationarity by reducing sensitivity to per-instance scale and shift, thereby limiting systematic long-horizon error.

D. Additional analysis of DualMixFormer

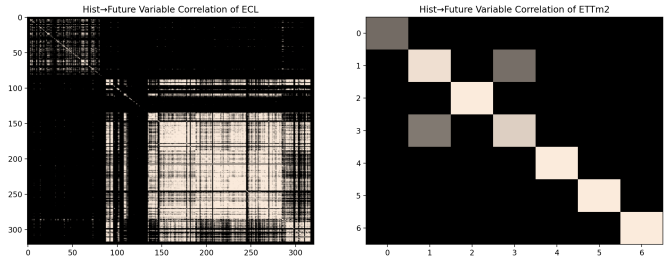
This section provides supplementary analyses to clarify why **DualMixFormer** performs well in long-horizon forecasting. We focus on three aspects: (i) cross-variable dependency patterns induced by attention, (ii) dataset-level correlation structures that indicate how much inter-variable signal is available, and (iii) robustness under missing observations. The interpretation follows the model's functional decomposition: a *lightweight MLP branch* provides a low-variance continuation bias for smooth trends and other regular components, while the *Transformer branch* captures nonlinear cross-variable residual interactions beyond simple extrapolation.

a) Why does DualMixFormer outperform purely linear models?: DualMixFormer improves on linear baselines in two key aspects. First, it explicitly models inter-variable dependencies using a Transformer encoder over variate tokens, allowing each channel to borrow information from others when beneficial. Second, the attention–FFN stack provides nonlinear capacity to capture higher-order interactions, so residual patterns that linear, trend-dominated continuation cannot represent can be corrected rather than accumulating as systematic error. Hence, the advantage should be larger when



(a) Electricity (b) ETTm2

Fig. 2. Inter-variable correlation in multivariate time series. Electricity exhibits stronger correlations than ETTm2.



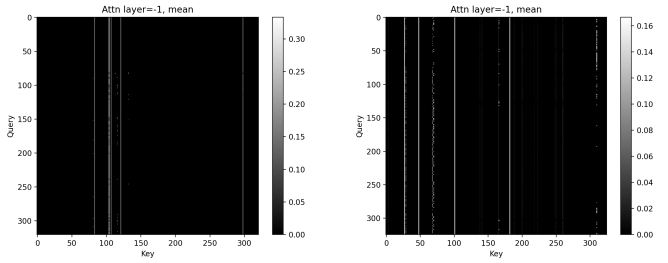
(a) Electricity (b) ETTm2

Fig. 3. Similarity between historical and future variables. Electricity shows stronger historical–future associations than ETTm2.

cross-variable coupling is strong and residual dynamics deviate from linear extrapolation.

To verify this, we analyze correlation structures on **ETTm2** and **Electricity** with a unified protocol. The look-back length is fixed to 336, and 100 training sub-series segments of length 336 are randomly sampled. For each segment, we compute the Pearson correlation matrix across variables and average the 100 matrices. For visualization, a common threshold $\tau = 0.5$ is applied to $|\rho|$ to highlight entries with $|\rho| \geq \tau$. As shown in Fig. 2, Electricity exhibits clear block-wise high-correlation regions, whereas ETTm2 is sparser with strong associations limited to a subset of variables. This indicates that Electricity offers richer cross-variable signals that can complement within-channel extrapolation; when correlations are weak, the benefit of cross-variable correction naturally diminishes.

We further examine **historical–future** correlations to assess cross-variable predictability over the forecast horizon. For each segment $\{X_{t:t+335}\}$, we compute $\rho_{ij} = \mathrm{corr}(X_{t:t+335,i},\, X_{t+336:t+671,j})$ between the historical window and the subsequent future window, then average over the same 100 samples and visualize with $\tau = 0.5$. Fig. 3 shows that Electricity retains clear historical–future associations, suggesting that the history of some variables informs the future of others, while ETTm2 displays much weaker structure. This gap helps explain why cross-variable modeling yields larger gains on Electricity-like benchmarks.



(a) MEVA-conditioned attention (b) Standard attention

Fig. 4. Visualization of attention matrices from the last encoder layer averaged over heads.

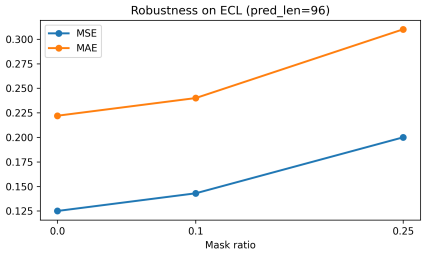
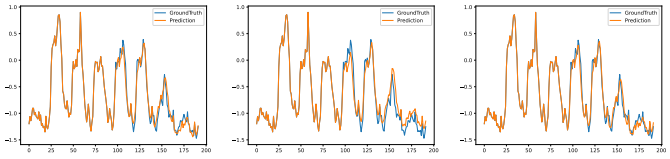


Fig. 5. Robustness under random masking on the Electricity dataset with prediction horizon T=96.

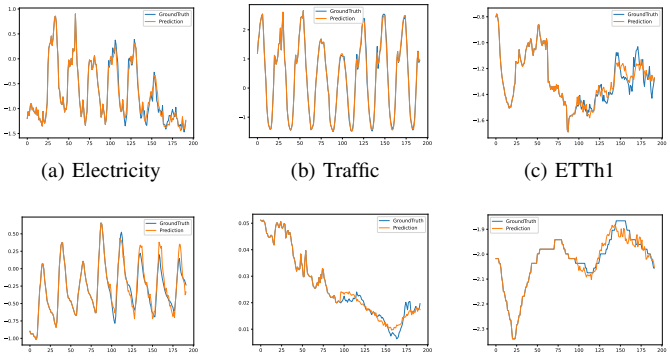
b) Visualization of cross-variable attention matrices.: Fig. 4 shows attention matrices from the last encoder layer, averaged across heads. Under variate tokenization, attention weights represent an explicit routing policy over variables rather than temporal alignment. Diffuse attention would indicate weak selectivity and be less useful when the Transformer acts as a residual corrector. Instead, attention mass concentrates on a small set of high-weight columns, implying that many query variables repeatedly attend to a subset of hub variables. With MEVA conditioning, the allocation becomes sharper—fewer columns are activated while peak weights increase—suggesting that identity and phase cues reduce ambiguity and help isolate salient cross-variable residual links. This aligns with the fusion principle: the MLP branch provides stable smooth continuation, while attention selectively adjusts residual deviations driven by inter-variable structure.

c) Robustness to missing observations.: Robustness is evaluated by randomly masking the historical input with ratio r and filling masked entries with zeros, while keeping targets unchanged. Results on Electricity are shown in Fig. 5. Degradation is moderate under mild missingness (e.g., $r = 0.1$) but increases as r grows (e.g., $r = 0.25$), as the effective information in the context window is reduced. The hybrid design remains useful under partial observation: the MLP branch still provides a stable continuation bias, while the Transformer branch can aggregate residual evidence from the remaining observed variables when dependencies persist. Severe missingness remains challenging for long-horizon forecasting and is left for future work.



(a) DualMixFormer (b) PatchTST (c) iTransformer

Fig. 6. Prediction comparison on the Electricity dataset. DualMixFormer preserves finer temporal details.



(a) Electricity (b) Traffic (c) ETTh1

(d) ETTh2 (e) Weather (f) ETTm2

Fig. 7. Prediction performance of DualMixFormer across multiple datasets.

V. Prediction Showcases

Fig. 6 qualitatively compares model outputs on identical inputs, while Fig. 7 shows representative **DualMixFormer** forecasts across datasets. DualMixFormer tracks local fluctuations more faithfully than competing methods, aligning with the intended hybrid design: the *lightweight MLP branch* provides a low-variance continuation of smooth trend and regular components, whereas the Transformer branch performs cross-variable residual corrections to preserve fine-grained deviations without amplifying long-horizon error.

The visualizations also indicate that improvements are dataset-dependent. When dynamics exhibit weak inter-variable coupling, strong noise, or pronounced non-stationarity, cross-variable information for residual modeling is limited; attention-based correction becomes less effective and the forecast relies more on the smooth-oriented MLP pathway. Thus, while the fusion strategy improves robustness by combining smooth continuation with residual "repair," further gains remain possible on datasets where stochasticity or distributional drift diminishes cross-variable utility.

VI. Conclusion

This work proposes a *variable-centric* framework for multivariate long-term time series forecasting, motivated by three common challenges in long-horizon prediction: (i) temporal-dominant attention often underutilizes inter-variable dependencies, (ii) variate tokenization weakens explicit periodic phase cues, and (iii) predictive stability degrades as the horizon increases. To address these issues, we redefine self-attention as *cross-variable* aggregation by treating variables as tokens, thereby enforcing explicit inter-channel modeling and direct multivariate information fusion.

We further introduce MEVA to condition inverted variable tokens with **channel** and **phase** embeddings. Channel embeddings provide stable variable identity priors, while phase embeddings inject periodic-stage context that is otherwise obscured after tokenization. With this explicit conditioning, the Transformer branch is primarily used to capture nonlinear cross-variable residual structure, including long-range and higher-order interactions.

The proposed model employs an encoder-only cross-variable Transformer with a lightweight projection head for horizon generation, avoiding decoder complexity. To improve long-horizon robustness, an integrated lightweight MLP branch provides a low-variance continuation bias for smooth trends and quasi-linear extrapolation; its prediction is fused with the attention branch via learnable channel-wise weights, yielding a clear division of labor: smooth continuation by the MLP and residual correction by attention. RevIN is adopted to mitigate instance-wise distribution shift and scale variability. Experiments on multiple benchmarks demonstrate strong performance, with ablations validating the contributions of MEVA, the MLP branch, and RevIN. Interpretability analyses further reveal how cross-variable attention allocates dependency mass across channels and how dataset correlation structure constrains residual correction.

References

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, Hoboken, NJ, USA: Wiley, 2015.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/NECO.1997.9.8.1735.

[3] P. Hewage *et al.*, "Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station," *Soft Comput.*, vol. 24, pp. 16453–16482, 2020, doi: 10.1007/s00500-020-04954-0.

[4] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecast.*, vol. 36, no. 3, pp. 1181–1191, 2020, doi: 10.1016/j.ijforecast.2019.07.001.

[5] R. Sen, H.-F. Yu, and I. S. Dhillon, "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 4837–4846, 2019.

[6] T. Zhang *et al.*, "Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures," *arXiv preprint* arXiv:2207.01186, 2022.

[7] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[8] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, pp. 11106–11115, 2021, doi: 10.1609/AAAI.V35I12.17325.

[9] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 22419–22430, 2021.

[10] T. Zhou *et al.*, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, ser. *Proc. Mach. Learn. Res.*, vol. 162. PMLR, pp. 27268–27286, 2022.

[11] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, pp. 11121–11128, 2023, doi: 10.1609/AAAI.V37I9.26317.

[12] Y. Liu *et al.*, "iTransformer: Inverted transformers are effective for time series forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

[13] T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible instance normalization for accurate time-series forecasting against distribution shift," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2022.

[14] V. Cerqueira, N. Moniz, and C. Soares, "VEST: Automatic feature engineering for forecasting," *Mach. Learn.*, vol. 113, pp. 4523–4545, 2024, doi: 10.1007/s10994-021-05959-y.

[15] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, 1991, doi: 10.1016/0925-2312(91)90023-5.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[17] P. Cui, Z. Deng, W. Hu, and J. Zhu, "SDE-HNN: Accurate and well-calibrated forecasting using stochastic differential equations," *ACM Trans. Knowl. Discov. Data*, vol. 19, no. 2, pp. 1–23, 2025, doi: 10.1145/3691346.

[18] Y. Liu, P. Cui, W. Hu, and R. Hong, "Deep ensembles meets quantile regression: Uncertainty-aware imputation for time series," *arXiv preprint* arXiv:2312.01294, 2023.

[19] P. Cui, W. Hu, and J. Zhu, "Calibrated reliable regression using maximum mean discrepancy," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 17164–17175, 2020.

[20] Z. Zhang, W. Hu, T. Tian, and J. Zhu, "Dynamic window-level granger causality of multi-channel time series," *arXiv preprint* arXiv:2006.07788, 2020.

[21] M. Liu *et al.*, "SCINet: Time series modeling and forecasting with sample convolution and interaction," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 5816–5828, 2022.

[22] Q. Pan, W. Hu, and N. Chen, "Two birds with one stone: Series saliency for accurate and interpretable multivariate time series forecasting," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 2884–2891, 2021.

[23] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, pp. 95–104, 2018, doi: 10.1145/3209978.3210006.

[24] Z. Wu *et al.*, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, pp. 753–763, 2020, doi: 10.1145/3394486.3403118.

[25] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, pp. 1421–1441, 2019, doi: 10.1007/s10994-019-05815-0.

[26] H. Wu *et al.*, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT)*, pp. 4171–4186, 2019, doi: 10.18653/V1/N19-1423.

[28] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877–1901, 2020.

[29] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint* arXiv:2005.08100, 2020.

[30] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 10012–10022, 2021, doi: 10.1109/ICCV48922.2021.00986.

[31] S. Li *et al.*, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[32] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 9881–9893, 2022.

[33] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[34] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.

[35] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021.

[36] S. Lin *et al.*, "CycleNet: Enhancing time series forecasting through modeling periodic patterns," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.