Gateformer: Advancing Multivariate Time Series Forecasting via Temporal and Variate-Wise Attention with Gated Representations

Yu-Hsiang Lan [1]

Abstract

Transformer-based models have recently shown promise in time series forecasting, yet effectively modeling multivariate time series remains challenging due to the need to capture both temporal (cross-time) and variate (cross-variate) dependencies. While prior methods attempt to address both, it remains unclear how to optimally integrate these dependencies within the Transformer architecture for both accuracy and efficiency. We re-purpose the Transformer to explicitly model these two types of dependencies: first embedding each variate independently to capture temporal dynamics, then applying attention over these embeddings to model cross-variate relationships. Gating mechanisms in both stages regulate information flow, enabling the model to focus on relevant features. Our approach achieves state-of-the-art performance on 13 real-world datasets and can be integrated into Transformer-, LLM-based, and foundation time series models, improving performance by up to 20.7%. Code is available at this repository: https://github.com/nyuolab/Gateformer.

1. Introduction

Time series forecasting plays a critical role in domains such as traffic (Lv et al., 2014), energy (Zhu et al., 2023), weather (Angryk et al., 2020), healthcare (Kaushik et al., 2020), and finance (Chen et al., 2012). In multivariate settings, accurate forecasting depends on modeling both intra-series (temporal) and inter-series (cross-variable) dependencies (Zhang & Yan, 2023). Transformers (Vaswani et al., 2017), originally successful in NLP, have been adapted for time series due to their ability to model long-range dependencies (Wen et al., 2023).

[1]New York University, New York, NY, USA. Correspondence to: Yu-Hsiang Lan <yl12081@nyu.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

Many Transformer-based models (Kitaev et al., 2020; Li et al., 2021; Liu et al., 2022b) embed all variables at each time step into a single token and apply attention across time. However, these tokens often lack semantic depth, weakening temporal modeling—sometimes allowing simple linear models to outperform them (Zeng et al., 2023). PatchTST (Nie et al., 2023) addresses this by using patch-level temporal attention but overlooks cross-variable dependencies. iTransformer (Liu et al., 2024b) models cross-variable correlations via coarse embeddings, sacrificing fine-grained temporal detail.

We propose **Gateformer**, a Transformer-based model that explicitly captures both temporal and cross-variate dependencies. Each variate is first encoded independently to capture temporal patterns, then integrated with global temporal context through a gating mechanism that regulates information flow. These representations serve as inputs for cross-variate modeling.

To model cross-variate dependencies, we apply self-attention over variate representations, producing variate-interacting embeddings. While such modeling improves capacity (Han et al., 2024), it may degrade performance on low-dimensional datasets. To address this, we fuse variate-interacting and non-interacting embeddings through a second gating mechanism, dynamically controlling cross-variate influence and ensuring robust performance across dataset scales. Our main contributions in this work can be summarized as follows:

- We introduce Gateformer, which combines temporal and variate-wise attention with gated representations to improve multivariate time series forecasting.
- Gateformer achieves state-of-the-art results on 13 real-world benchmarks, ranking top-1 in 91 and top-2 in 122 out of 130 settings.
- Our propsed framework seamlessly integrates with Transformer-based, LLM-based, and foundation time series models, achieving performance improvements of up to 20.7% and facilitating the development of foundational multivariate models.

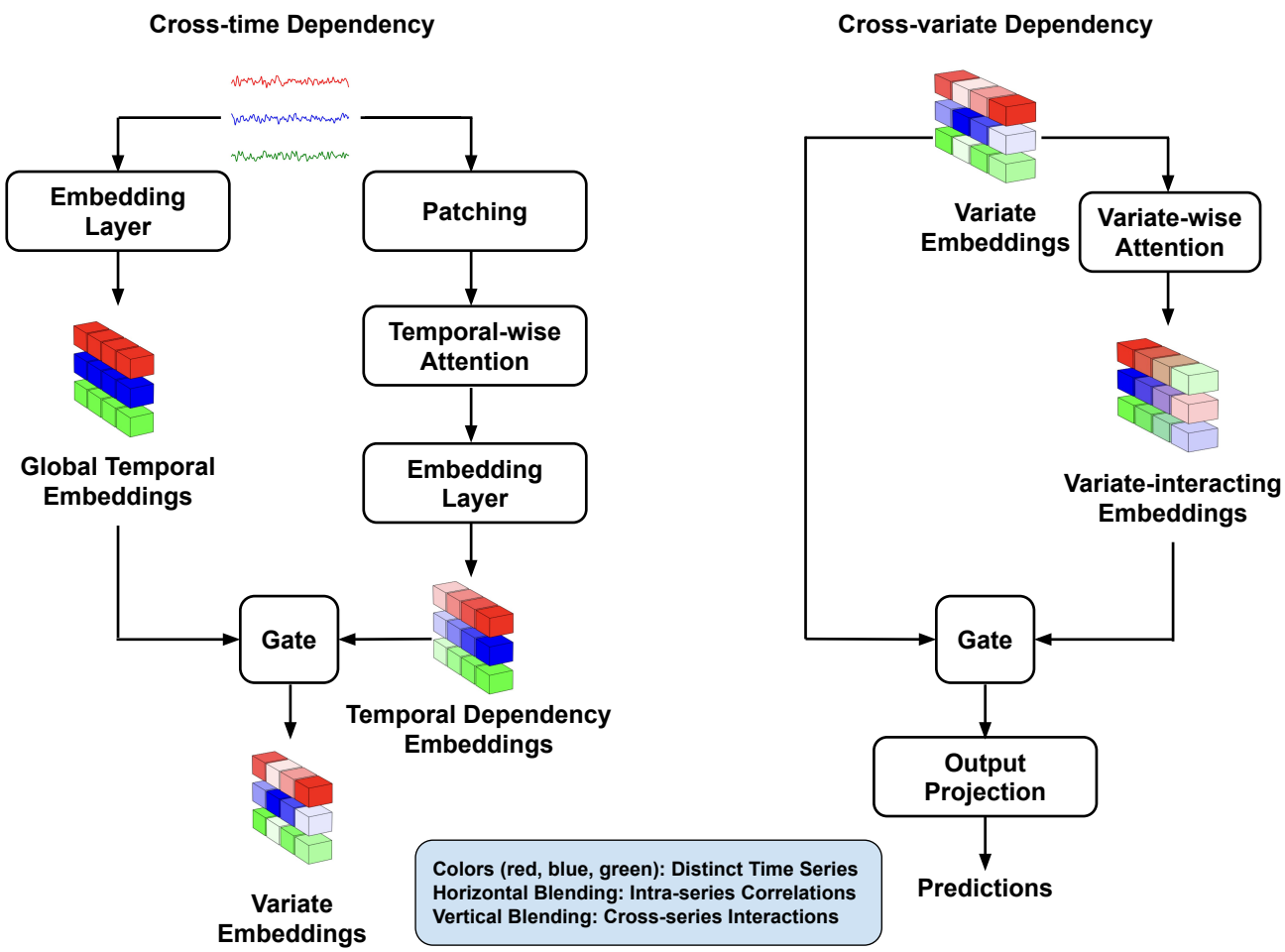

Figure 1. Overall model architecture: The model encodes each variate's series independently through two distinct pathways to obtain variate-wise representations: (1) temporal dependency embeddings that capture cross-time dependencies through patching and temporal-wise attention, and (2) global temporal embeddings that encode global temporal patterns through an MLP. These complementary embeddings are integrated through a gating operation to form variate embeddings, which serve as input for cross-variate dependency modeling. Variate-wise attention is then applied on variate embeddings to model multivariate dependencies, producing variate-interacting embeddings. Finally, a copy of the variate embeddings (without interaction) is combined with the variate-interacting embeddings through gating to regulate cross-variate correlations. The resulting output is then passed through a projection layer to generate predictions.

2. Methodology

2.1. Problem Definition

Given multivariate time series data $\mathbf{X} \in \mathbb{R}^{N \times T}$, the goal is to forecast the next F steps, $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times F}$, where N is the number of variates and T is the look-back window. The objective is to minimize the MSE between $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$.

2.2. Model Overview

As shown in Figure 1, our model captures both temporal and cross-variate dependencies.

Cross-time Dependency Modeling. Each variate is processed independently to capture intra-series patterns. Two parallel paths are used: (1) patch-based temporal attention to capture local dependencies, and (2) a shared MLP to learn global temporal patterns. These are fused via a gating mechanism to form variate embeddings.

Cross-variate Dependency Modeling. Variate embeddings are passed through a variate-wise attention module to model inter-series relationships. To maintain performance on small datasets, these outputs are gated with original variate embeddings to dynamically control cross-variate influence.

Output Projection. Final embeddings are passed through a linear layer to produce predictions $\hat{\mathbf{Y}}$, followed by denormalization. MSE is used for training.

Table 1. Multivariate forecasting results with prediction lengths $T \in \{96, 192, 336, 720\}$ and fixed look-back length $L = 96$. Results are averaged from all prediction lengths. A lower value indicates better performance. Full results are listed in the Appendix J.

Models	**Gateformer (Ours)**		iTransformer (2024b)		PatchTST (2023)		Crossformer (2023)		FEDformer (2022)		Autoformer (2021)		Stationary (2022c)		TimesNet (2023)		SCINet (2022a)		DLinear (2023)	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT(Avg)	**0.373**	**0.394**	0.383	0.399	0.381	0.397	0.685	0.578	0.408	0.428	0.465	0.459	0.471	0.464	0.391	0.404	0.689	0.597	0.442	0.444
Weather	**0.253**	**0.276**	0.258	0.278	0.259	0.281	0.259	0.315	0.309	0.360	0.338	0.382	0.288	0.314	0.259	0.287	0.292	0.363	0.265	0.317
Electricity	**0.176**	**0.267**	0.178	0.270	0.205	0.290	0.244	0.334	0.214	0.327	0.227	0.338	0.193	0.296	0.192	0.295	0.268	0.365	0.212	0.300
Traffic	**0.412**	**0.276**	0.428	0.282	0.481	0.304	0.550	0.304	0.610	0.376	0.628	0.379	0.624	0.340	0.620	0.336	0.804	0.509	0.625	0.383
Exchange	**0.326**	**0.386**	0.360	0.403	0.367	0.404	0.940	0.707	0.519	0.429	0.613	0.539	0.461	0.454	0.416	0.443	0.750	0.626	0.354	0.414
Solar-Energy	**0.224**	**0.258**	0.233	0.262	0.270	0.307	0.641	0.639	0.291	0.381	0.885	0.711	0.261	0.381	0.301	0.319	0.282	0.375	0.330	0.401
PEMS(Avg)	**0.070**	**0.169**	0.072	0.173	0.104	0.211	0.103	0.195	0.137	0.253	0.346	0.431	0.089	0.193	0.091	0.193	0.072	0.174	0.133	0.252
1st Count	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2. Ablation study of model components across eight datasets, using prediction length $T = 96$ and look-back window $L = 96$.

Models	ETTh1		ETTm1		ETTh2		ETTm2		Exchange		Electricity		Traffic		Weather		Avg. rank	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Gateformer	0.383	0.398	**0.320**	**0.360**	**0.306**	**0.351**	**0.176**	**0.260**	**0.081**	**0.199**	0.146	**0.238**	0.390	**0.261**	**0.168**	**0.208**	**1.375**	**1.125**
w/o Temporal-wise Attn.	0.393	0.404	0.327	0.361	0.325	0.360	0.179	0.261	0.084	0.203	0.147	0.239	0.395	0.266	0.176	0.217	3.25	3.000
w/o Gate in Variate-wise Attn.	0.385	0.400	0.327	0.362	0.334	0.367	0.182	0.264	0.082	0.200	**0.144**	0.239	**0.387**	**0.261**	**0.168**	**0.208**	2.250	2.375
w/o Global Temporal Embeddings	**0.381**	**0.397**	0.323	0.362	0.343	0.367	0.179	0.262	0.084	0.200	0.146	0.241	0.392	**0.261**	0.172	0.212	2.750	2.500

2.3. Model Details

2.3.1. Cross-time Dependency Modeling

Each series $\mathbf{x}^{(i)} \in \mathbb{R}^T$ is normalized with RevIN (Kim et al., 2022), then divided into P non-overlapping patches. Each patch is projected to a d_m-dimensional space and passed through self-attention:

$$\mathbf{O}^{(i)}_{cross-time} = \text{SOFTMAX}\left(\frac{\mathbf{Q}^{(i)}\mathbf{K}^{(i)\top}}{\sqrt{d_k}}\right)\mathbf{V}^{(i)}$$

The attention output is reshaped and passed through an FFN to obtain temporal embeddings $\mathbf{v}^{(i)}_T \in \mathbb{R}^{d_m}$.

Global Temporal Pattern. To address the loss of global context from patching, each raw series is also processed by a shared MLP to obtain a global embedding $\mathbf{v}^{(i)}_G \in \mathbb{R}^{d_m}$.

2.3.2. Gated Fusion

We fuse $\mathbf{v}^{(i)}_T$ and $\mathbf{v}^{(i)}_G$ using a gate:

$$Gate = \sigma(\mathbf{v}^{(i)}_T \boldsymbol{W}_{g1} + \mathbf{v}^{(i)}_G \boldsymbol{W}_{g2})$$

$$\mathbf{s}^{(i)} = Gate \odot \mathbf{v}^{(i)}_T + (1 - Gate) \odot \mathbf{v}^{(i)}_G$$

$\mathbf{s}^{(i)}$ serves as the final variate embedding.

2.3.3. Cross-variate Dependency Modeling

Given all variate embeddings $\mathbf{S} \in \mathbb{R}^{N \times d_m}$, we apply self-attention:

$$\mathbf{O}_{cross-variate} = \text{SOFTMAX}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_m}}\right)\mathbf{V}$$

To ensure robustness across dataset sizes, the output is gated with the original $\mathbf{S}$ embeddings.

2.3.4. Output Projection Layer

We apply a linear head on $\mathbf{O}_{cross-variate}$ to obtain final predictions $\hat{\mathbf{Y}} \in \mathbb{R}^{N \times F}$. Predictions are de-normalized using stored mean and standard deviation, and MSE is used as the training loss.

3. Experiments

We evaluate our model on short- and long-term forecasting tasks against nine SOTA methods. Results show strong performance and notable gains when applied to Transformer- and LLM-based forecasters. Ablation studies highlight the contribution of each component.

Datasets. We use 13 real-world datasets, including Traffic, Electricity, Weather, Exchange, four ETT subsets (Wu et al., 2021), Solar-Energy (Lai et al., 2018), and four PEMS subsets (Liu et al., 2022a). Dataset details are provided in Appendix B.1.

3.1. Forecasting Results

Baselines We compare our model against nine SOTA methods: Transformer-based (Autoformer, FEDformer, Stationary, Crossformer, PatchTST, iTransformer), linear-based (DLinear), and TCN-based (SCINet, TimesNet).

Experimental Setup All models use a look-back window of $L = 96$ and are trained for 10 epochs. Pre-

Table 3. Full results of performance improvements achieved using our proposed framework.

Models			Autoformer (2021)		Flowformer (2022)		GPT4TS (2023)		Moment (2024)	
Metric			MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	Original	96	0.201	0.317	0.215	0.320	0.197	0.290	0.204	0.296
		192	0.222	0.334	0.259	0.355	0.201	0.292	0.207	0.299
		336	0.231	0.338	0.296	0.383	0.217	0.309	0.219	0.310
		720	0.254	0.361	0.296	0.380	0.253	0.339	0.256	0.341
		Avg	0.227	0.338	0.267	0.359	0.217	0.308	0.221	0.311
	+ Our framework	96	**0.173**	**0.259**	**0.144**	**0.237**	**0.150**	**0.241**	**0.163**	**0.254**
		192	**0.180**	**0.267**	**0.160**	**0.251**	**0.163**	**0.254**	**0.177**	**0.269**
		336	**0.198**	**0.285**	**0.175**	**0.269**	**0.180**	**0.272**	**0.195**	**0.283**
		720	**0.241**	**0.320**	**0.205**	**0.296**	**0.214**	**0.301**	**0.226**	**0.311**
		Avg	**0.198**	**0.283**	**0.187**	**0.274**	**0.177**	**0.267**	**0.190**	**0.279**
Weather	Original	96	0.266	0.336	0.182	0.233	0.203	0.244	0.192	0.234
		192	0.307	0.367	0.250	0.288	0.247	0.277	0.246	0.278
		336	0.359	0.395	0.309	0.329	0.297	0.311	0.287	0.305
		720	0.419	0.428	0.404	0.385	0.368	0.356	0.360	0.350
		Avg	0.338	0.382	0.286	0.308	0.279	0.297	0.271	0.292
	+ Our framework	96	**0.175**	**0.218**	**0.171**	**0.213**	**0.175**	**0.214**	**0.184**	**0.223**
		192	**0.219**	**0.258**	**0.220**	**0.258**	**0.228**	**0.260**	**0.225**	**0.259**
		336	**0.281**	**0.300**	**0.275**	**0.297**	**0.282**	**0.301**	**0.281**	**0.301**
		720	**0.353**	**0.350**	**0.353**	**0.349**	**0.359**	**0.352**	**0.356**	**0.349**
		Avg	**0.257**	**0.281**	**0.255**	**0.279**	**0.261**	**0.282**	**0.261**	**0.283**

diction lengths are $T \in \{3, 6, 12, 24\}$ for PEMS, and $T \in \{96, 192, 336, 720\}$ for others. Baseline results follow Liu et al. (2024b); PEMS results for $T = 12, 24$ are also from Liu et al. (2024b). Experiments are implemented in PyTorch (Paszke et al., 2019) and run on a single NVIDIA A100 GPU. Baselines are reproduced using the TimesNet (Wu et al., 2023) repository.

Main Results Table 1 shows that Gateformer consistently outperforms all baselines across 13 datasets. Although iTransformer and Crossformer capture cross-variate dependencies, they fall short due to limitations in modeling temporal dynamics or introducing noise through patch-wise interactions. Our method encodes both temporal and global patterns via gated representations, enabling robust cross-variate modeling. This flexibility allows strong performance across dataset scales. On low-dimensional datasets (e.g., ETT), Gateformer matches or exceeds PatchTST, a strong channel-independent baseline, thanks to its gated control of cross-variate interactions.

3.2. Model Analysis

Ablation Study We assess component contributions on eight datasets (ETT, Weather, Traffic, Electricity, Exchange) by removing: (1) **Temporal-wise Attention** (keeps only global patterns), (2) **Gate in Variate-wise Attention**, and (3) **Global Temporal Embeddings** (keeps only local patterns).

Table 2 shows full Gateformer achieves the best results. Removing temporal attention significantly degrades performance, especially on small datasets, confirming its importance. Removing the variate-wise gate slightly improves performance on large datasets but harms it on smaller ones. Overall, the gated integration of temporal and variate-wise dependencies ensures strong, consistent forecasting performance.

3.3. Framework Generalizability

While capturing cross-variate dependencies is essential, the quadratic cost of self-attention limits scalability with many variates. Our Transformer-based framework addresses this by easily integrating efficient attention modules from models like Autoformer (Wu et al., 2021) and Flowformer (Wu et al., 2022), and by extending to LLM-based models like GPT4TS (Zhou et al., 2023) or foundation models such as Moment (Goswami et al., 2024). On Weather and Electricity datasets, our framework improves Autoformer by 19.9%, Flowformer by 20.7%, GPT4TS by 10.8% and Moment by 6.9%, as shown in Table 3. These gains highlight the flexibility and effectiveness of our approach.

4. Conclusion and Future Work

We present a model that captures both temporal and cross-variate dependencies using attention and gating mechanisms for improved forecasting. It achieves state-of-the-art results and boosts performance of existing Transformer-based models by up to 20.7%. Future work includes large-scale pre-training and broader time series applications.

Software and Data

Code, dataset details, metrics, and experimental settings are included in Appendix B.

References

Angryk, R. A., Martens, P. C., Aydin, B., Kempton, D., Mahajan, S. S., Basodi, S., Ahmadzadeh, A., Cai, X., Filali Boubrahimi, S., Hamdi, S. M., et al. Multivariate time series dataset for space weather data analytics. Scientific data, 7(1):227, 2020.

Chen, C. W., Gerlach, R., Lin, E. M., and Lee, W. Bayesian forecasting for financial risk management, pre and post the global financial crisis. Journal of Forecasting, 31(8): 661–687, 2012.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In Forty-first International Conference on Machine Learning, 2024. URL https://openreview.net/forum?id=jn2iTJas6h.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Han, L., Ye, H.-J., and Zhan, D.-C. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. IEEE Transactions on Knowledge and Data Engineering, 2024.

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. Time-LLM: Time series forecasting by reprogramming large language models. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Kaushik, S., Choudhury, A., Sheron, P. K., Dasgupta, N., Natarajan, S., Pickett, L. A., and Dutt, V. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. Frontiers in big data, 3:4, 2020.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Kitaev, N., Kaiser, L., and Levskaya, A. Reformer: The efficient transformer. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long- and short-term temporal patterns with deep neural networks, 2018. URL https://arxiv.org/abs/1703.07015.

Li, J., Hui, X., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. arXiv: 2012.07436, 2021.

Liu, M., Zeng, A., Chen, M., Xu, Z., LAI, Q., Ma, L., and Xu, Q. SCINet: Time series modeling and forecasting with sample convolution and interaction. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022a. URL https://openreview.net/forum?id=AyajSjTAzmg.

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., and Dustdar, S. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In International Conference on Learning Representations, 2022b. URL https://openreview.net/forum?id=0EXmFzUn5I.

Liu, X., Hu, J., Li, Y., Diao, S., Liang, Y., Hooi, B., and Zimmermann, R. Unitime: A language-empowered unified model for cross-domain time series forecasting. In The Web Conference 2024, 2024a. URL https://openreview.net/forum?id=P6sKyx2xAB.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in Neural Information Processing Systems, 35:9881–9893, 2022c.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. In The Twelfth International Conference on Learning Representations, 2024b. URL https://openreview.net/forum?id=JePfAI8fah.

Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. Timer: Generative pre-trained transformers are large time series models. In Forty-first International

Conference on Machine Learning, 2024c. URL https://openreview.net/forum?id=bYRYb7DMNo.

Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F.-Y. Traffic flow prediction with big data: A deep learning approach. Ieee transactions on intelligent transportation systems, 16(2):865–873, 2014.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Advances in Neural Information Processing Systems, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. Transformers in time series: A survey, 2023. URL https://arxiv.org/abs/2202.07125.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=J4gRj6d5Qm.

Wu, H., Wu, J., Xu, J., Wang, J., and Long, M. Flowformer: Linearizing transformers with conservation flows. arXiv preprint arXiv:2202.06258, 2022.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In Proceedings of the AAAI conference on artificial intelligence, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317/26089.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In The Eleventh International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022. URL https://arxiv.org/abs/2201.12740.

Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained LM. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=gMS6FVZvmF.

Zhu, Z., Chen, W., Xia, R., Zhou, T., Niu, P., Peng, B., Wang, W., Liu, H., Ma, Z., Gu, X., et al. Energy forecasting with robust, flexible, and explainable machine learning algorithms. AI Magazine, 44(4):377–393, 2023.

A. Related work

As shown in Figure 2, Transformer-based time series models can be categorized based on their granularity of representations (point-wise, patch-wise, and variate-wise) and their approach to modeling cross-time and cross-variate dependencies.

Point-wise Representations with Cross-time Attention
Many prior works embed multiple variates at the same timestamp into point tokens and apply attention mechanisms to capture temporal dependencies among them. Examples include Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), and Pyraformer (Liu et al., 2022b), which focus on optimizing the quadratic complexity of self-attention. Autoformer detects sub-series similarities with $O(L \log L)$ complexity, FEDformer leverages frequency-domain sparsity with $O(L)$ complexity, and Pyraformer uses pyramidal attention for short- and long-term dependencies with $O(L)$ complexity. However, these models often fail in multivariate forecasting due to the lack of explicit semantic meaning in point tokens and the loss of cross-variate correlations when merging all variables into single temporal tokens.

Patch-wise Representations with Cross-time Attention PatchTST (Nie et al., 2023) addresses the lack of local semantic information through patching. It divides time series into patches to expand the receptive field and applies self-attention to capture cross-time dependencies among patches. PatchTST processes each variate independently, allowing unique attention patterns to be learned for each series. This approach has consistently improved performance on benchmarks and inspired recent large-scale time series models (Zhou et al., 2023; Das et al., 2024; Jin et al., 2024; Liu et al., 2024c). However, it focuses primarily on cross-time dependencies while neglecting critical cross-variate interactions.

Variate-wise Representations with Cross-variate Attention A notable example is iTransformer (Liu et al., 2024b), which expands the receptive field as an extreme form of patching. iTransformer encodes each variate's time series into coarse variate-wise representations through linear projection and explicitly models cross-variate correlations among these representations. However, it does not account for cross-time (intra-series) dependencies.

Patch-wise Representations with Cross-time and Cross-variate Attention Crossformer (Zhang & Yan, 2023), a representative model, segments each variate's series into a sequence of patches and captures both cross-time (intra-series) and cross-variate (inter-series) dependencies among them. However, modeling multivariate correlations at such a granular level can introduce unnecessary noise, potentially resulting in suboptimal forecasting performance.

Unlike previous works, our approach captures both cross-time and cross-variate correlations through variate-wise representations. To enhance the quality of these representations, we integrate gating mechanisms to regulate information flow, allowing the model to focus on the most relevant features for accurate predictions.

B. Experimental Details

B.1. Datasets

We assess long-term forecasting performance on ten widely used benchmarks, including the four ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), Weather, Electricity, Traffic, Exchange (from Wu et al. (2021)), and Solar-Energy (from Lai et al. (2018)). Additionally, we evaluate short-term forecasting on the four PEMS datasets (PEMS03, PEMS04, PEMS07, PEMS08) as used in Liu et al. (2022a). We adopt the same data processing and train-validation-test split protocol as TimesNet (Wu et al., 2023), ensuring datasets are strictly divided chronologically to prevent data leakage. For forecasting, the lookback series length is fixed at $L = 96$ across all datasets. Prediction lengths are set as follows: $T \in \{96, 192, 336, 720\}$ for ETT, Weather, ECL, Traffic, Solar-Energy, and Exchange; $T \in \{3, 6, 12, 24\}$ for PEMS. Detailed dataset information is provided in Table 4.

B.2. Implementation Details

All experiments were implemented in PyTorch (Paszke et al., 2019) and conducted on a single NVIDIA A100 40GB GPU. We used MSE as the loss function, a batch size of 8, and the Adam optimizer. The initial learning rate was selected from $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, with training running for a maximum of 10 epochs and early stopping applied if validation loss did not improve within three epochs. The number of encoder blocks was chosen from $L \in \{1, 2, 3, 4\}$, and the hidden state dimension was selected from $\{64, 128, 256, 512, 1024\}$. Evaluation metrics included mean square error (MSE) and mean absolute error (MAE), with results averaged over three random seeds. Baseline models were reproduced using the TimesNet (Wu et al., 2023) repository, following configurations from their respective original papers or official code.

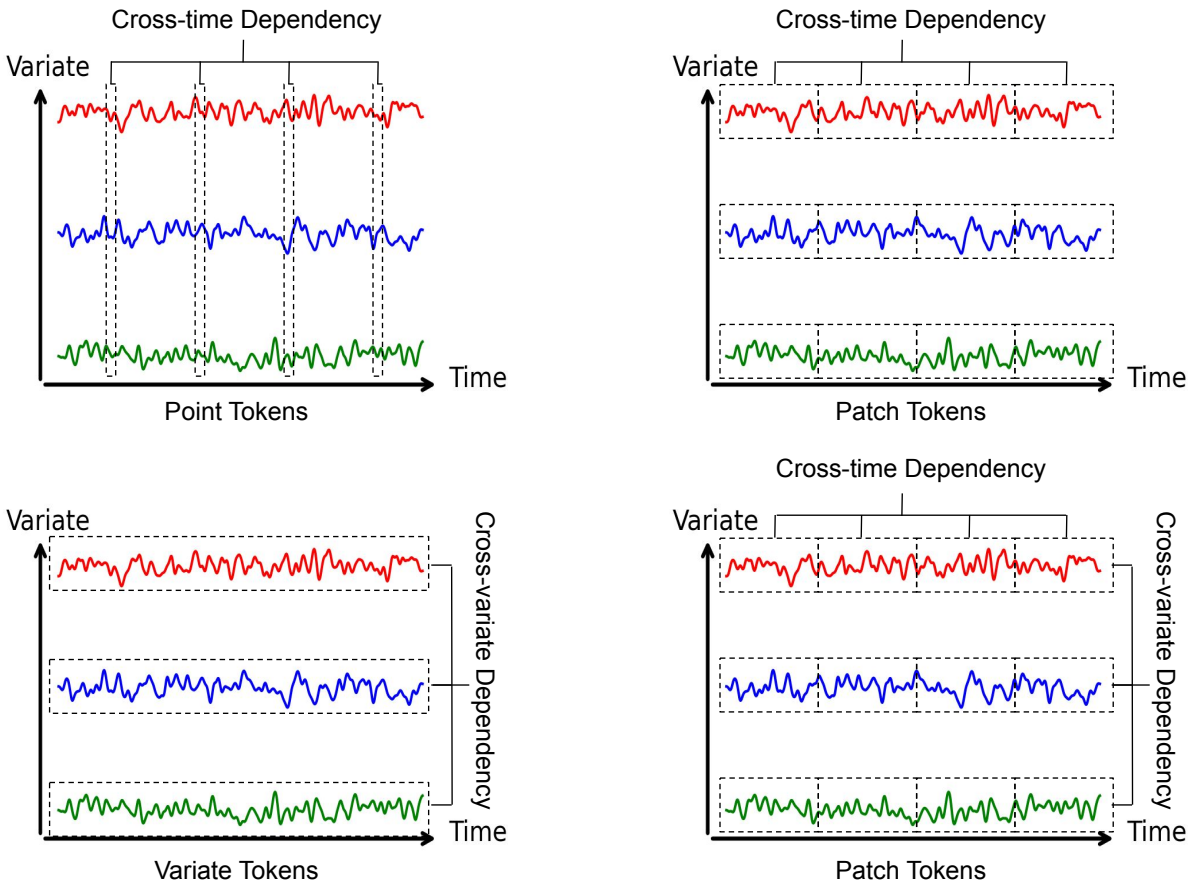

Figure 2. Transformer-based forecasters can be categorized based on their attention granularity (point-wise, patch-wise, and variate-wise) and their approach to modeling cross-time and cross-variate dependencies.

Table 4. Detailed dataset descriptions. The dimension refers to the number of variates in each dataset and the dataset size is organized in (training, validation, testing).

Dataset	Dim.	Prediction Length	Dataset Size	Frequency	Information
ETTh1, ETTh2	7	{96, 192, 336, 720}	(8545, 2881, 2881)	Hourly	Electricity
ETTm1, ETTm2	7	{96, 192, 336, 720}	(34465, 11521, 11521)	15min	Electricity
Exchange	8	{96, 192, 336, 720}	(5120, 665, 1422)	Daily	Economy
Weather	21	{96, 192, 336, 720}	(36792, 5271, 10540)	10min	Weather
ECL	321	{96, 192, 336, 720}	(18317, 2633, 5261)	Hourly	Electricity
Traffic	862	{96, 192, 336, 720}	(12185, 1757, 3509)	Hourly	Transportation
Solar-Energy	137	{96, 192, 336, 720}	(36601, 5161, 10417)	10min	Energy
PEMS03	358	{3, 6, 12, 24}	(15617, 5135, 5135)	5min	Transportation
PEMS04	307	{3, 6, 12, 24}	(10172, 3375, 3375)	5min	Transportation
PEMS07	883	{3, 6, 12, 24}	(16911, 5622, 5622)	5min	Transportation
PEMS08	170	{3, 6, 12, 24}	(10690, 3548, 3548)	5min	Transportation

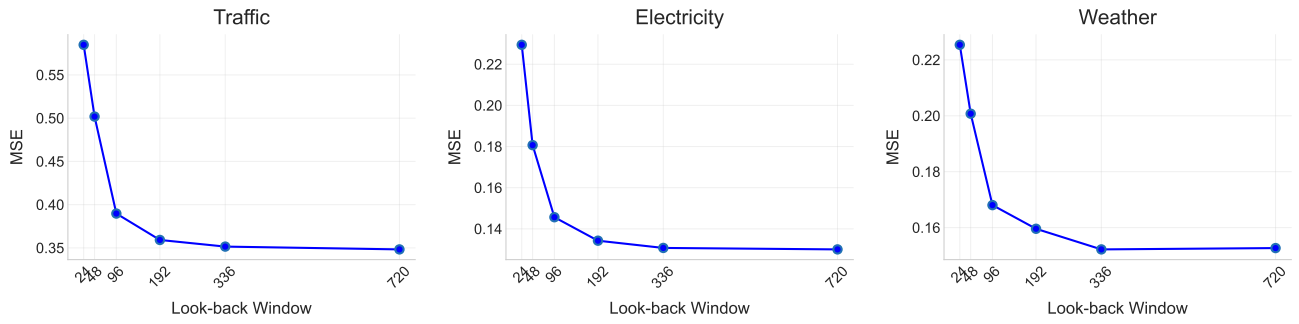

Figure 3. Forecasting performance (MSE) across different look-back windows $L \in \{24, 48, 96, 192, 336, 720\}$ on Traffic, Electricity, and Weather datasets, with prediction length $T = 96$.

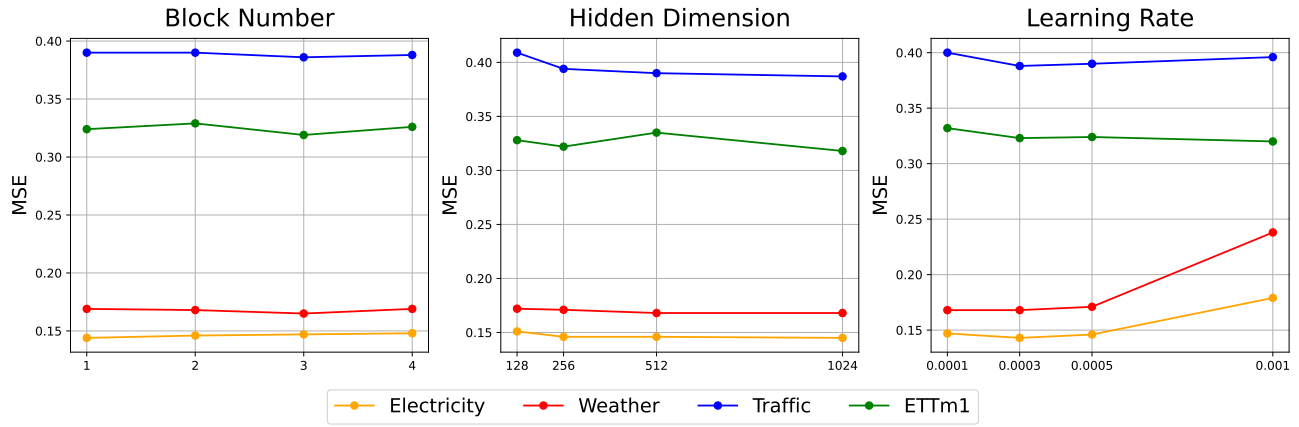

Figure 4. Forecasting results (MSE) varying with the number of Transformer blocks, the model's hidden dimension, and the learning rate. The results were collected with a prediction horizon of $T = 96$ and a look-back window of $L = 96$.

C. Varying Look-back Window

In principle, expanding the look-back window increases the receptive field, which should enhance forecasting accuracy. However, previous studies (Zeng et al., 2023) have shown that most Transformer-based models fail to demonstrate this expected improvement, revealing their limitations in processing extended temporal sequences. Our model, in contrast, effectively utilizes the increased receptive field, achieving lower MSE scores with longer look-back windows, as demonstrated in Figure 3.

D. Hyperparameter Sensitivity

To evaluate the sensitivity of our method to hyperparameter settings, we conducted experiments by varying the number of Transformer blocks $L \in \{1, 2, 3, 4\}$, model dimensions $D \in \{128, 256, 512, 1024\}$, and learning rates $lr \in \{10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ on the ETTm1, Weather, Electricity, and Traffic datasets. As shown in Figure 4, the learning rate has the strongest impact on model performance and requires careful tuning. Model performance generally improves with larger hidden dimensions, while remaining relatively stable across different numbers of Transformer blocks.

E. Framework Generalizability

As discussed in Section 3.3, our proposed framework can be seamlessly integrated into other Transformer-based and LLM-based models to consistently improve their performance, with full forecasting results shown in Table 3. Since our model explicitly captures cross-variate dependencies, the variate-wise attention becomes a computational bottleneck when training on high-dimensional datasets such as Electricity, Traffic, and Solar-Energy. To mitigate this, we replace the conventional quadratic-time attention mechanism with the linear-time attention proposed in Flowformer (Wu et al., 2022) and compare its performance against competitive baselines. These results are provided in Figure 5.

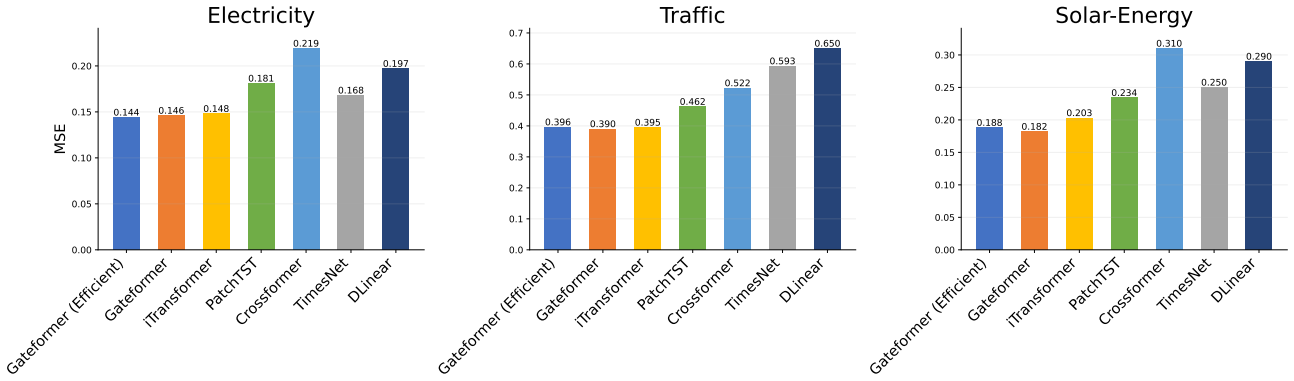

Figure 5. Performance comparison on high-dimensional datasets. Gateformer (Efficient) replaces the conventional quadratic-time attention mechanism with Flowformer's optimized linear-time attention to improve computational efficiency. Prediction horizon $T = 96$; look-back window $L = 96$.

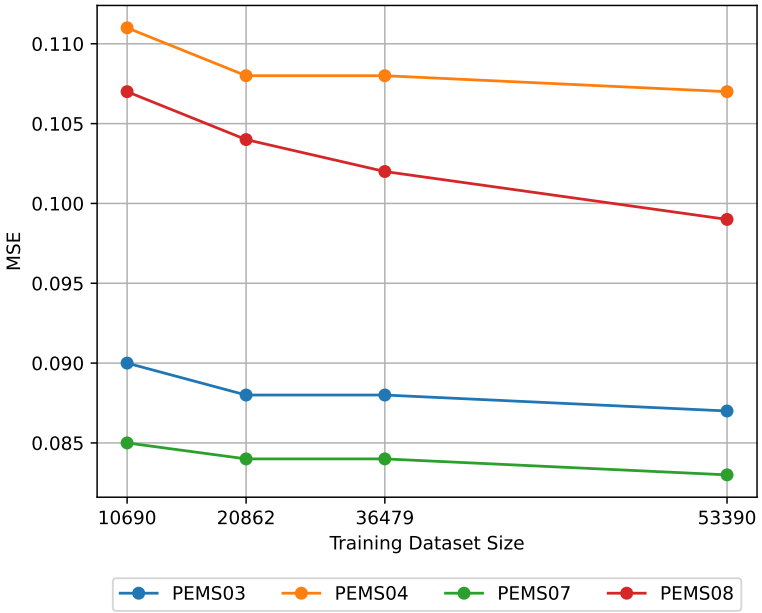

Figure 6. Analysis of unified cross-dataset training: Models are pre-trained on four combinations of PEMS datasets with increasing training set sizes, then fine-tuned on target datasets. Prediction length $T = 24$; look-back window $L = 96$.

F. Unified Cross-Dataset Training

We selected the PEMS dataset as our benchmark as it contains four high-dimensional time-series subsets (PEMS03, PEMS04, PEMS07, and PEMS08), each with substantial data that ensures robust evaluation. Our model includes temporal-wise and variate-wise self-attention modules. The temporal-wise attention module captures intra-series dependencies for each series independently. The variate-wise attention module operates on the variate dimension, enabling unified training across datasets with varying numbers of variates. To investigate the benefits of data scaling, we trained our model on four progressively larger combined datasets: (1) PEMS08, (2) PEMS08 + PEMS04, (3) PEMS08 + PEMS04 + PEMS03, and (4) PEMS08 + PEMS04 + PEMS03 + PEMS07. These datasets were combined in ascending order of dimensionality. As shown in Figure 6, forecasting accuracy improved consistently across all PEMS datasets with larger training set sizes. This highlights our model's ability to learn generalizable and transferable representations through unified training while preventing catastrophic forgetting, a common challenge in combined training (Liu et al., 2024a). Our results highlight the potential of building large-scale multivariate forecasting models trained across datasets using our framework.

Table 5. Transfer learning results. Gateformer* is pre-trained on PEMS07 dataset and then transferred to other datasets. Best results are marked in **bold**, with prediction lengths $T \in \{3, 6, 12, 24\}$ and look-back window $L = 96$.

Models		Gateformer*				Models Trained from Scratch on Target Dataset									
		Zero-shot		Fine-tuning		Gateformer		iTransformer		PatchTST		Crossformer		FEDformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	3	0.049	0.148	**0.044**	**0.140**	0.045	0.142	0.047	0.147	0.055	0.166	0.051	0.150	0.109	0.231
	6	0.064	0.167	**0.052**	**0.152**	0.053	0.153	0.057	0.159	0.069	0.184	0.060	0.161	0.112	0.235
	12	0.092	0.198	**0.065**	**0.168**	0.066	0.170	0.071	0.174	0.099	0.216	0.090	0.203	0.126	0.251
	24	0.146	0.251	**0.088**	**0.197**	0.092	0.201	0.093	0.201	0.142	0.259	0.121	0.240	0.149	0.275
	Avg	0.088	0.191	**0.062**	**0.164**	0.064	0.166	0.067	0.170	0.091	0.206	0.081	0.189	0.124	0.248
PEMS04	3	0.064	0.164	**0.060**	**0.156**	0.061	0.159	0.064	0.164	0.071	0.184	0.062	0.162	0.122	0.250
	6	0.077	0.181	**0.067**	**0.166**	0.068	0.168	0.073	0.175	0.081	0.191	0.069	0.173	0.119	0.245
	12	0.101	0.210	0.081	**0.183**	0.083	0.186	**0.078**	**0.183**	0.105	0.224	0.098	0.218	0.138	0.262
	24	0.162	0.269	0.110	0.216	0.114	0.221	**0.095**	**0.205**	0.153	0.275	0.131	0.256	0.177	0.293
	Avg	0.101	0.206	0.080	0.180	0.081	0.184	**0.078**	**0.182**	0.103	0.219	0.090	0.202	0.139	0.263
PEMS08	3	0.057	0.155	**0.051**	**0.144**	0.052	0.147	0.055	0.153	0.064	0.175	0.117	0.161	0.153	0.255
	6	0.070	0.172	**0.059**	**0.154**	0.060	0.156	0.064	0.165	0.076	0.190	0.129	0.173	0.157	0.257
	12	0.094	0.200	**0.073**	**0.170**	0.075	0.172	0.079	0.182	0.168	0.232	0.165	0.214	0.173	0.273
	24	0.157	0.258	**0.104**	**0.200**	0.108	0.204	0.115	0.219	0.224	0.281	0.215	0.260	0.210	0.301
	Avg	0.094	0.196	**0.072**	**0.167**	0.074	0.170	0.078	0.180	0.133	0.220	0.157	0.202	0.173	0.272

G. REPRESENTATION LEARNING

We evaluate the representations learned by our model, focusing on their generalization and transferability across datasets. The PEMS datasets are used as a benchmark, as they comprise four high-dimensional time series subsets, each containing a substantial amount of data to ensure robust and reliable evaluation results. To assess representation transferability, we pre-trained the model on the largest dataset (PEMS07) for 10 epochs, then transferring it to PEMS03, PEMS04, and PEMS08 datasets. As shown in Table 5, the model's zero-shot forecasting performance is comparable to most competitive baseline models. A key strength of our method lies in its ability to effectively capture cross-variate correlations, enabling superior generalization and transferability to unseen datasets. Furthermore, pre-training followed by fine-tuning for a few epochs consistently outperforms training solely on downstream datasets, demonstrating the value of transferable representations learned during pre-training.

H. MODEL EFFICIENCY

H.1. Memory-Efficient Training Strategy

Our model captures multivariate correlations by applying the attention mechanism along the variate dimension. However, the quadratic complexity of the attention mechanism becomes a computational bottleneck when training models on high-dimensional datasets such as Traffic. To mitigate this, we implement a computationally efficient training strategy that randomly samples a subset of variates for each batch, allowing the model to train on only the selected variates. During inference, the model generates forecasts for all variates. This random sampling of variates acts as a form of regularization, enabling the model to learn robust and generalizable patterns. As shown in Figure 7, the forecasting performance remains stable across various sampling ratios, while the memory footprint is significantly reduced.

H.2. Model Efficiency

To evaluate the efficiency of our model, we compare the forecasting accuracy, training time, and memory footprint of the following models: Gateformer, Gateformer with the memory-efficient training strategy, iTransformer (Liu et al., 2024b), PatchTST (Nie et al., 2023), Crossformer (Zhang & Yan, 2023), TimesNet (Wu et al., 2023), and DLinear (Zeng et al., 2023). For a fair comparison, all models use the same hidden dimension and batch size on a representative high-dimensional dataset (Traffic).

As shown in Figure 8, in terms of training speed, the linear model (DLinear) is the fastest. Gateformer, which captures both cross-time and cross-variate dependencies, is slower than iTransformer, which models only multivariate correlations, and PatchTST, which focuses purely on cross-time correlations. However, by adopting an efficient training strategy (training on a

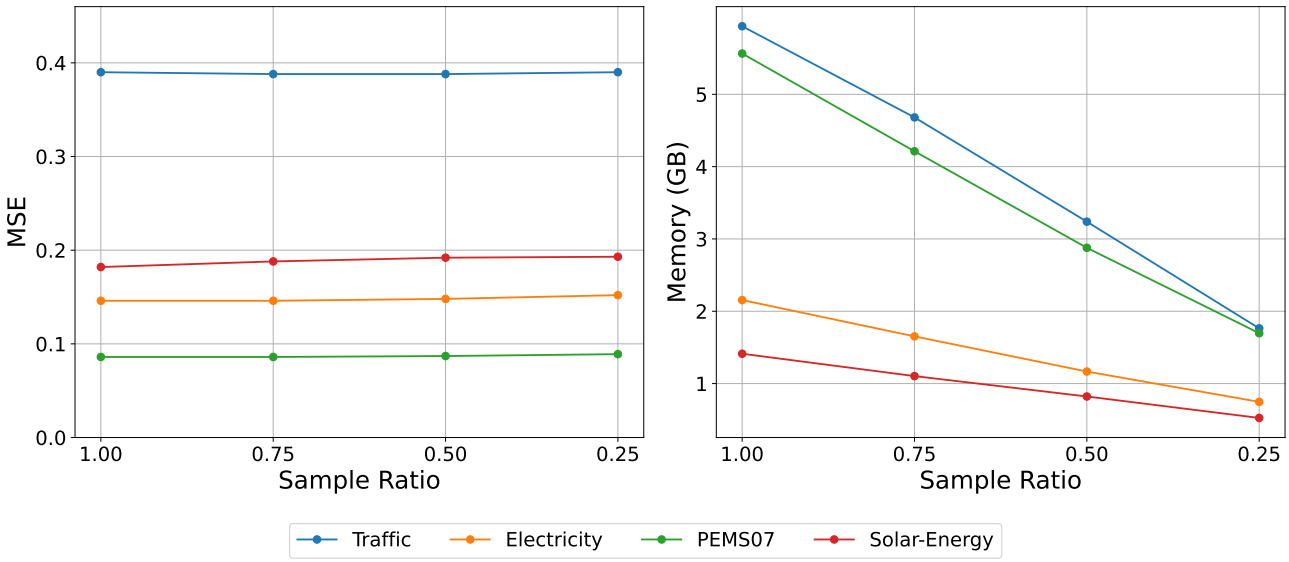

Figure 7. Investigation of the memory-efficient training strategy: The memory footprint (right) is significantly reduced when training on randomly selected variates, while maintaining consistent results (left) across different sampling ratios.

randomly sampled 20% of variates in each batch while forecasting all variates), Gateformer (Efficient) achieves comparable training speed to iTransformer with a lower memory footprint and superior forecasting performance.

I. Visualization

We visualize long-term forecasting results of Gateformer against baseline models on the Traffic dataset in Figure 9, where Gateformer demonstrates superior prediction accuracy. Figure 10 compares the forecasting performance of Transformer-based models with and without integration of our framework on the Electricity dataset. Models integrated with our approach consistently show improved prediction accuracy.

J. Full Multivariate Forecasting Results

We extensively evaluated our model's performance against competitive baselines on well-acknowledged forecasting benchmarks. Table 6 presents comprehensive short-term forecasting results on the four PEMS subsets. Table 7 details long-term forecasting results across nine challenging benchmarks.

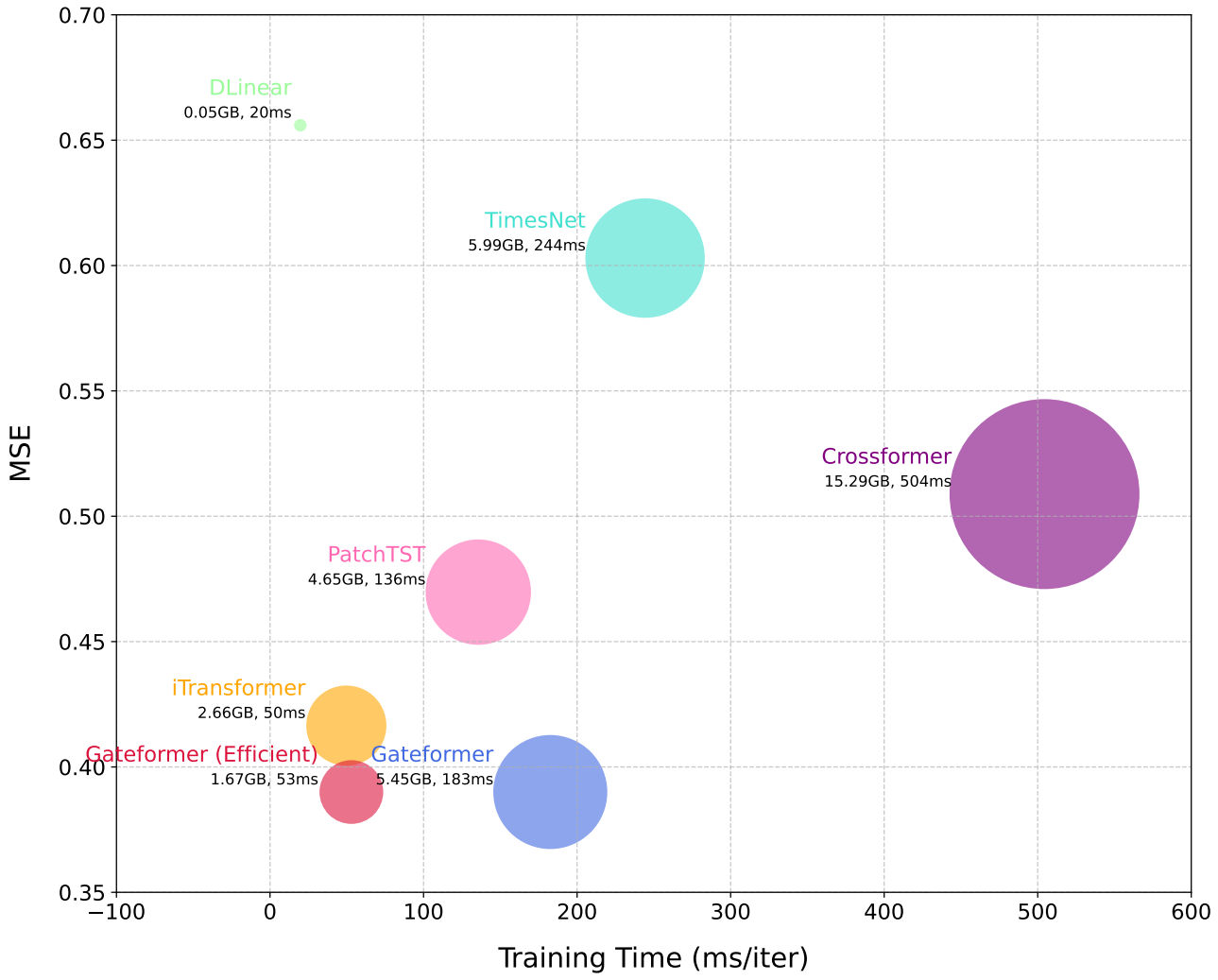

Figure 8. Model efficiency comparison on Traffic dataset.

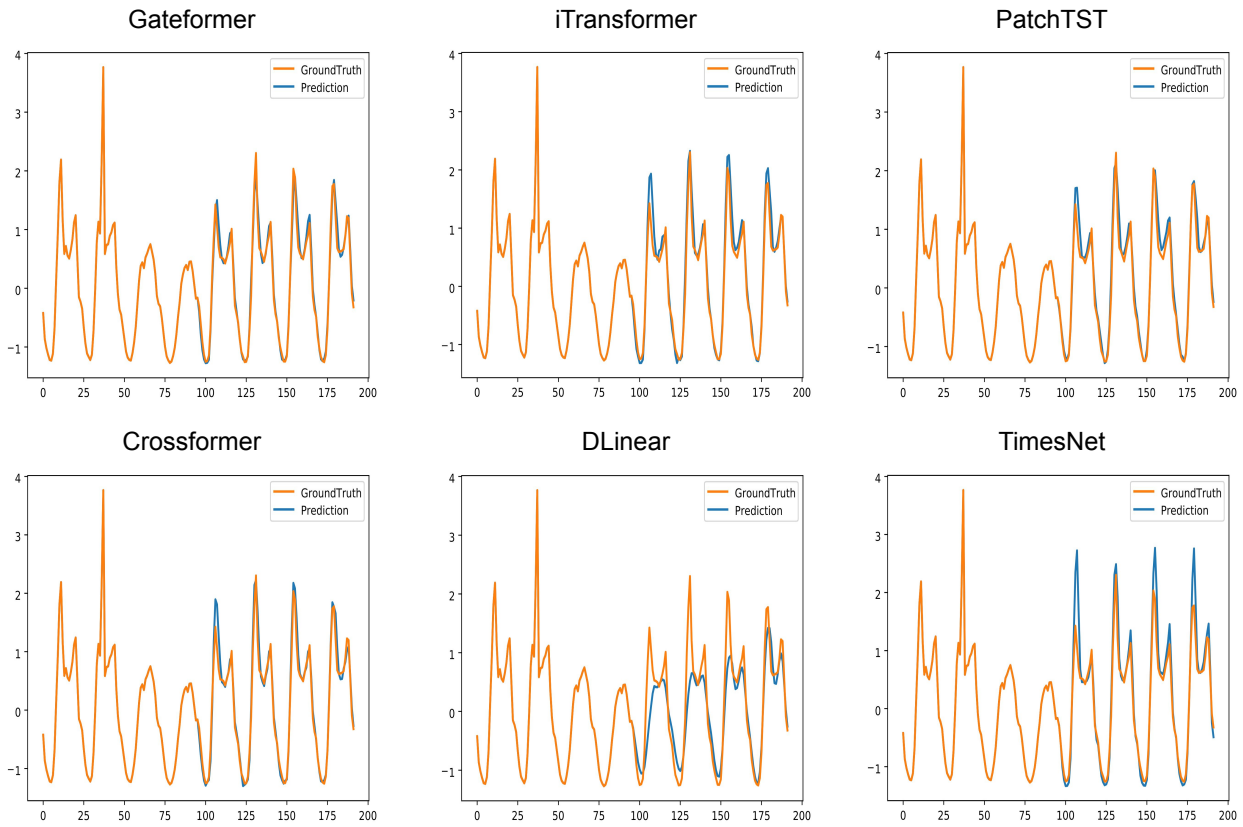

Figure 9. Visualization of 96-step forecasting on Traffic dataset with look-back window $L = 96$.

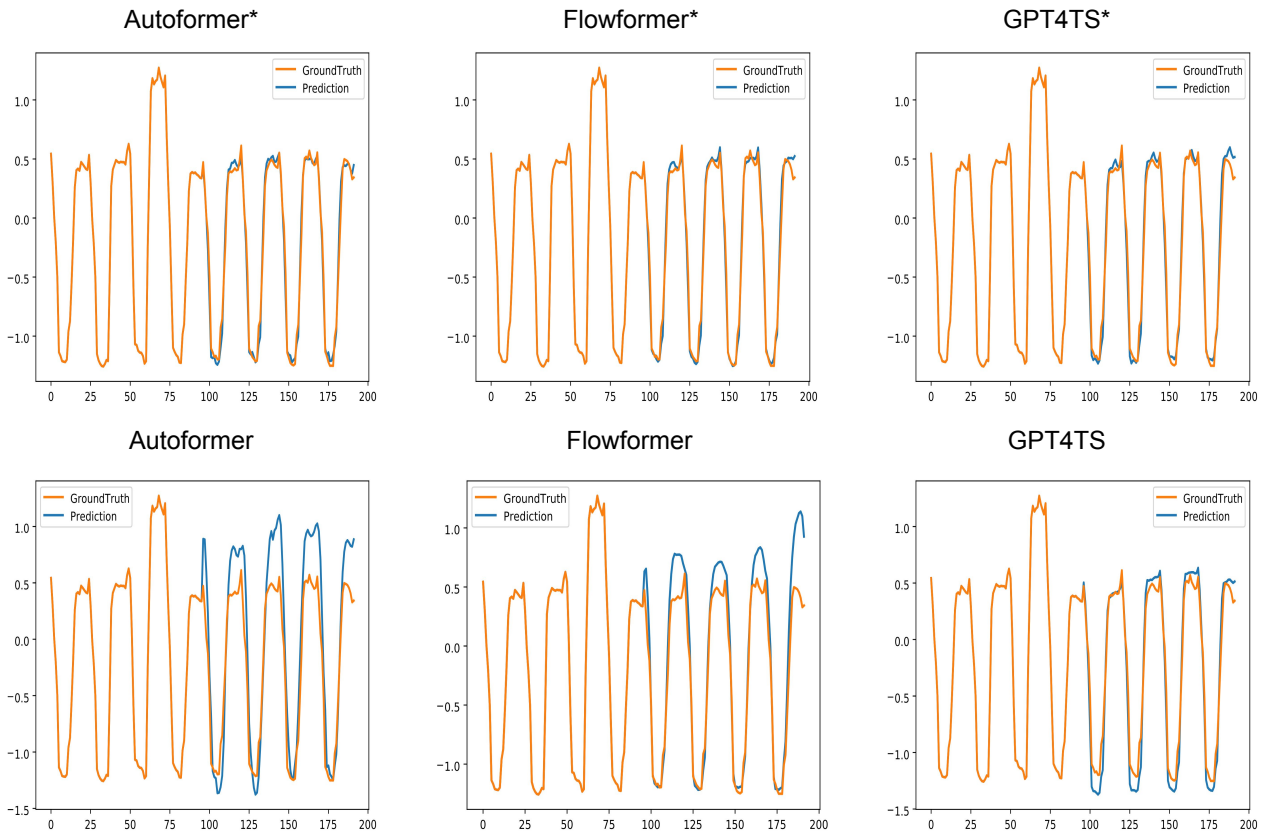

Figure 10. Visualization of 96-step forecasting on Electricity dataset with look-back window $L = 96$. * denotes models integrated with our framework.

Table 6. Full short-term multivariate forecasting results with prediction lengths $T \in \{3, 6, 12, 24\}$ and input length $L = 96$. The best results are highlighted in **red** and the second bests are marked in blue.

Model		**Gateformer** (**Ours**)		iTransformer (2024b)		PatchTST (2023)		Crossformer (2023)		FEDformer (2022)		Autoformer (2021)		Stationary (2022c)		TimesNet (2023)		SCINet (2022a)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PEMS03	3	**0.045**	**0.142**	0.047	0.147	0.055	0.166	0.051	0.150	0.109	0.231	0.285	0.394	0.063	0.168	0.068	0.173	0.048	0.147	0.069	0.183
	6	**0.053**	**0.153**	0.057	0.159	0.069	0.184	0.060	0.161	0.112	0.235	0.254	0.363	0.070	0.176	0.074	0.179	0.056	0.158	0.086	0.204
	12	**0.066**	**0.170**	0.071	0.174	0.099	0.216	0.090	0.203	0.126	0.251	0.272	0.385	0.081	0.188	0.085	0.192	**0.066**	0.172	0.122	0.243
	24	0.092	0.201	0.093	0.201	0.142	0.259	0.121	0.240	0.149	0.275	0.334	0.440	0.105	0.214	0.118	0.223	**0.085**	**0.198**	0.201	0.317
	Avg	**0.064**	**0.166**	0.067	0.170	0.091	0.206	0.081	0.189	0.124	0.248	0.286	0.396	0.080	0.187	0.086	0.192	**0.064**	0.169	0.120	0.237
PEMS04	3	0.061	**0.159**	0.064	0.164	0.071	0.184	0.062	0.162	0.122	0.250	0.305	0.414	0.076	0.181	0.075	0.179	**0.060**	**0.159**	0.096	0.218
	6	0.068	**0.168**	0.073	0.175	0.081	0.191	0.069	0.173	0.119	0.245	0.361	0.449	0.080	0.187	0.079	0.183	**0.067**	0.169	0.113	0.236
	12	0.083	0.186	0.078	0.183	0.105	0.224	0.098	0.218	0.138	0.262	0.424	0.491	0.088	0.196	0.087	0.195	**0.073**	**0.177**	0.148	0.272
	24	0.114	0.221	0.095	0.205	0.153	0.275	0.131	0.256	0.177	0.293	0.459	0.509	0.104	0.216	0.103	0.215	**0.084**	**0.193**	0.224	0.340
	Avg	0.081	0.184	0.078	0.182	0.103	0.219	0.090	0.202	0.139	0.263	0.387	0.466	0.087	0.195	0.086	0.193	**0.071**	**0.175**	0.145	0.267
PEMS07	3	**0.042**	**0.130**	0.046	0.139	0.052	0.158	0.050	0.141	0.102	0.218	0.201	0.326	0.069	0.169	0.068	0.165	0.043	0.132	0.061	0.172
	6	**0.049**	**0.141**	0.054	0.150	0.061	0.169	0.057	0.152	0.104	0.219	0.253	0.373	0.074	0.175	0.073	0.171	0.050	0.144	0.078	0.196
	12	**0.061**	**0.156**	0.067	0.165	0.095	0.207	0.094	0.200	0.109	0.225	0.199	0.336	0.083	0.185	0.082	0.181	0.068	0.171	0.115	0.242
	24	**0.086**	**0.187**	0.088	0.190	0.150	0.262	0.139	0.247	0.125	0.244	0.323	0.420	0.102	0.207	0.101	0.204	0.119	0.225	0.210	0.329
	Avg	**0.060**	**0.154**	0.064	0.161	0.090	0.199	0.085	0.185	0.110	0.227	0.244	0.364	0.082	0.184	0.081	0.180	0.070	0.168	0.116	0.235
PEMS08	3	**0.052**	**0.147**	0.055	0.153	0.064	0.175	0.117	0.161	0.153	0.255	0.434	0.463	0.084	0.183	0.088	0.186	0.059	0.158	0.094	0.215
	6	**0.060**	**0.156**	0.064	0.165	0.076	0.190	0.129	0.173	0.157	0.257	0.526	0.541	0.092	0.191	0.096	0.195	0.065	0.167	0.112	0.234
	12	**0.075**	**0.172**	0.079	0.182	0.168	0.232	0.165	0.214	0.173	0.273	0.436	0.485	0.109	0.207	0.112	0.212	0.087	0.184	0.154	0.276
	24	**0.108**	**0.204**	0.115	0.219	0.224	0.281	0.215	0.260	0.210	0.301	0.467	0.502	0.140	0.236	0.141	0.238	0.122	0.221	0.248	0.353
	Avg	**0.074**	**0.170**	0.078	0.180	0.133	0.220	0.157	0.202	0.173	0.272	0.466	0.498	0.106	0.204	0.109	0.208	0.083	0.183	0.152	0.270
	1st Count	**14**	**16**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	5	0	0

Table 7. Full results of multivariate long-term forecasting with a fixed input length $L = 96$ for all datasets. Baseline results are from Liu et al. (2024b). The best results are highlighted in **red** and the second bests are marked in blue.

Model		**Gateformer** (**Ours**)		iTransformer (2024b)		PatchTST (2023)		Crossformer (2023)		FEDformer (2022)		Autoformer (2021)		Stationary (2022c)		TimesNet (2023)		SCINet (2022a)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	**0.390**	**0.261**	0.395	0.268	0.462	0.295	0.522	0.290	0.587	0.366	0.613	0.388	0.612	0.338	0.593	0.321	0.788	0.499	0.650	0.396
	192	**0.409**	**0.270**	0.417	0.276	0.466	0.296	0.530	0.293	0.604	0.373	0.616	0.382	0.613	0.340	0.617	0.336	0.789	0.505	0.598	0.370
	336	**0.424**	**0.278**	0.433	0.283	0.482	0.304	0.558	0.305	0.621	0.383	0.622	0.337	0.618	0.328	0.629	0.336	0.797	0.508	0.605	0.373
	720	**0.459**	**0.296**	0.467	0.302	0.514	0.322	0.589	0.328	0.626	0.382	0.660	0.408	0.653	0.355	0.640	0.350	0.841	0.523	0.645	0.394
	Avg	**0.421**	**0.276**	0.428	0.282	0.481	0.304	0.550	0.304	0.610	0.376	0.628	0.379	0.624	0.340	0.620	0.336	0.804	0.509	0.625	0.383
Solar-Energy	96	**0.182**	**0.222**	0.203	0.237	0.234	0.286	0.310	0.331	0.242	0.342	0.884	0.711	0.215	0.249	0.250	0.292	0.237	0.344	0.290	0.378
	192	**0.227**	**0.257**	0.233	0.261	0.267	0.310	0.734	0.725	0.285	0.380	0.834	0.692	0.254	0.272	0.296	0.318	0.280	0.380	0.320	0.398
	336	**0.242**	0.274	0.248	**0.273**	0.290	0.315	0.750	0.735	0.282	0.376	0.941	0.723	0.290	0.296	0.319	0.330	0.304	0.389	0.353	0.415
	720	**0.245**	0.279	0.249	**0.275**	0.289	0.317	0.769	0.765	0.357	0.427	0.882	0.717	0.285	0.295	0.338	0.337	0.308	0.388	0.356	0.413
	Avg	**0.224**	**0.258**	0.233	0.262	0.270	0.307	0.641	0.639	0.291	0.381	0.885	0.711	0.261	0.381	0.301	0.319	0.282	0.375	0.330	0.401
Electricity	96	**0.146**	**0.238**	0.148	0.240	0.181	0.270	0.219	0.314	0.193	0.308	0.201	0.317	0.169	0.273	0.168	0.272	0.247	0.345	0.197	0.282
	192	**0.160**	**0.252**	0.162	0.253	0.188	0.274	0.231	0.322	0.201	0.315	0.222	0.334	0.182	0.286	0.184	0.289	0.257	0.355	0.196	0.285
	336	**0.176**	0.270	0.178	**0.269**	0.204	0.293	0.246	0.337	0.214	0.329	0.231	0.338	0.200	0.304	0.198	0.300	0.269	0.369	0.209	0.301
	720	0.221	**0.306**	0.225	0.317	0.246	0.324	0.280	0.363	0.246	0.355	0.254	0.361	0.222	0.321	**0.220**	0.320	0.299	0.390	0.245	0.333
	Avg	**0.176**	**0.267**	0.178	0.270	0.205	0.290	0.244	0.334	0.214	0.327	0.227	0.338	0.193	0.296	0.192	0.295	0.268	0.365	0.212	0.300
Weather	96	0.168	**0.208**	0.174	0.214	0.177	0.218	**0.158**	0.230	0.217	0.296	0.266	0.336	0.173	0.223	0.172	0.220	0.221	0.306	0.196	0.255
	192	0.216	**0.253**	0.221	0.254	0.225	0.259	**0.206**	0.277	0.276	0.336	0.307	0.367	0.245	0.285	0.219	0.261	0.261	0.340	0.237	0.296
	336	0.276	**0.296**	0.278	**0.296**	0.278	0.297	**0.272**	0.335	0.339	0.380	0.359	0.395	0.321	0.338	0.280	0.306	0.309	0.378	0.283	0.335
	720	0.352	0.348	0.358	**0.347**	0.354	0.348	0.398	0.418	0.403	0.428	0.419	0.428	0.414	0.410	0.365	0.359	0.377	0.427	**0.345**	0.381
	Avg	**0.253**	**0.276**	0.258	0.278	0.259	0.281	0.259	0.315	0.309	0.360	0.338	0.382	0.288	0.314	0.259	0.287	0.292	0.363	0.265	0.317
Exchange	96	**0.081**	**0.199**	0.086	0.206	0.088	0.205	0.256	0.367	0.148	0.278	0.197	0.323	0.111	0.237	0.107	0.234	0.267	0.396	0.088	0.218
	192	**0.167**	**0.293**	0.177	0.299	0.176	0.299	0.470	0.509	0.271	0.315	0.300	0.369	0.219	0.335	0.226	0.344	0.351	0.459	0.176	0.315
	336	0.312	0.403	0.331	0.417	**0.301**	**0.397**	1.268	0.883	0.460	0.427	0.509	0.524	0.421	0.476	0.367	0.448	1.324	0.853	0.313	0.427
	720	**0.744**	**0.650**	0.847	0.691	0.901	0.714	1.767	1.068	1.195	0.695	1.447	0.941	1.092	0.769	0.964	0.746	1.058	0.797	0.839	0.695
	Avg	**0.326**	**0.386**	0.360	0.403	0.367	0.404	0.940	0.707	0.519	0.429	0.613	0.539	0.461	0.454	0.416	0.443	0.750	0.626	0.354	0.414
ETTm1	96	**0.320**	**0.360**	0.334	0.368	0.329	0.367	0.404	0.426	0.379	0.419	0.505	0.475	0.386	0.398	0.338	0.375	0.418	0.438	0.345	0.372
	192	**0.367**	**0.383**	0.377	0.391	**0.367**	0.385	0.450	0.451	0.426	0.441	0.553	0.496	0.459	0.444	0.374	0.387	0.439	0.450	0.380	0.389
	336	0.406	0.411	0.426	0.420	**0.399**	**0.410**	0.532	0.515	0.445	0.459	0.621	0.537	0.495	0.464	0.410	0.411	0.490	0.485	0.413	0.413
	720	0459	0.447	0.491	0.459	**0.454**	**0.439**	0.666	0.589	0.543	0.490	0.671	0.561	0.585	0.516	0.478	0.450	0.595	0.550	0.474	0.453
	Avg	0.388	**0.400**	0.407	0.410	**0.387**	**0.400**	0.513	0.496	0.448	0.452	0.588	0.517	0.481	0.456	0.400	0.406	0.485	0.481	0.403	0.407
ETTm2	96	0.176	0.260	0.180	0.264	**0.175**	**0.259**	0.287	0.366	0.203	0.287	0.255	0.339	0.192	0.274	0.187	0.267	0.286	0.377	0.193	0.292
	192	0.245	0.307	0.250	0.309	**0.241**	**0.302**	0.414	0.492	0.269	0.328	0.281	0.340	0.280	0.339	0.249	0.309	0.399	0.445	0.284	0.362
	336	**0.305**	**0.343**	0.311	0.348	**0.305**	**0.343**	0.597	0.542	0.325	0.366	0.339	0.372	0.334	0.361	0.321	0.351	0.637	0.591	0.369	0.427
	720	0.404	**0.399**	0.412	0.407	**0.402**	0.400	1.730	1.042	0.421	0.415	0.433	0.432	0.417	0.413	0.408	0.403	0.960	0.735	0.554	0.522
	Avg	0.283	0.327	0.288	0.332	**0.281**	**0.326**	0.757	0.610	0.305	0.349	0.327	0.371	0.306	0.347	0.291	0.333	0.571	0.537	0.350	0.401
ETTh1	96	0.383	**0.398**	0.386	0.405	0.414	0.419	0.423	0.448	**0.376**	0.419	0.449	0.459	0.513	0.491	0.384	0.402	0.654	0.599	0.386	0.400
	192	0.433	0.435	0.441	0.436	0.460	0.445	0.471	0.474	**0.420**	0.448	0.500	0.482	0.534	0.504	0.436	**0.429**	0.719	0.631	0.437	0.432
	336	0.464	**0.452**	0.487	0.458	0.501	0.466	0.570	0.546	**0.459**	0.465	0.521	0.496	0.588	0.535	0.491	0.469	0.778	0.659	0.481	0.459
	720	**0.485**	**0.484**	0.503	0.491	0.500	0.488	0.653	0.621	0.506	0.507	0.514	0.512	0.643	0.616	0.521	0.500	0.836	0.699	0.519	0.516
	Avg	0.441	**0.442**	0.454	0.447	0.469	0.454	0.529	0.522	**0.440**	0.460	0.496	0.487	0.570	0.537	0.458	0.450	0.747	0.647	0.456	0.452
ETTh2	96	0.306	0.351	**0.297**	0.349	0.302	**0.348**	0.745	0.584	0.358	0.397	0.346	0.388	0.476	0.458	0.340	0.374	0.707	0.621	0.333	0.387
	192	**0.371**	**0.394**	0.380	0.400	0.388	0.400	0.877	0.656	0.429	0.439	0.456	0.452	0.512	0.493	0.402	0.414	0.860	0.689	0.477	0.476
	336	**0.415**	**0.430**	0.428	0.432	0.426	0.433	1.043	0.731	0.496	0.487	0.482	0.486	0.552	0.551	0.452	0.452	1.000	0.744	0.594	0.541
	720	**0.425**	**0.442**	0.427	0.445	0.431	0.446	1.104	0.763	0.463	0.474	0.515	0.511	0.562	0.560	0.462	0.468	1.249	0.838	0.831	0.657
	Avg	**0.379**	**0.404**	0.383	0.407	0.387	0.407	0.942	0.684	0.437	0.449	0.450	0.459	0.526	0.516	0.414	0.427	0.954	0.723	0.559	0.515
	1st Count	**28**	**33**	1	5	10	9	3	0	4	0	0	0	0	0	1	1	0	0	1	0