

Time Tracker: Mixture-of-Experts-Enhanced Foundation Time Series Forecasting Model with Decoupled Training Pipelines

Xiaohou Shi, Ke Li, Aobo Liang and Yan Sun

Abstract—In the past few years, time series foundation models have achieved superior predicting accuracy. However, real-world time series often exhibit significant diversity in their temporal patterns across different time spans and domains, making it challenging for a single model architecture to fit all complex scenarios. In addition, time series data may have multiple variables exhibiting complex correlations between each other. Recent mainstream works have focused on modeling times series in a channel-independent manner in both pretraining and finetuning stages, overlooking the valuable inter-series dependencies. To this end, we propose Time Tracker for better predictions on multivariate time series data. Firstly, we leverage sparse mixture of experts (MoE) within Transformers to handle the modeling of diverse time series patterns, thereby alleviating the learning difficulties of a single model while improving its generalization. Besides, we propose Any-variate Attention, enabling a unified model structure to seamlessly handle both univariate and multivariate time series, thereby supporting channel-independent modeling during pretraining and channel-mixed modeling for finetuning. Furthermore, we design a graph learning module that constructs relations among sequences from frequency-domain features, providing more precise guidance to capture inter-series dependencies in channel-mixed modeling. Based on these advancements, Time Tracker achieves state-of-the-art performance in predicting accuracy, model generalization and adaptability.

Index Terms—Time series forecasting, foundation model, mixture of experts, graph learning

I. INTRODUCTION

RECENT advances in time series forecasting have demonstrated the effectiveness of foundation models, with Transformers emerging as the pivotal architecture. With the auxiliary information of endogenous, exogenous variables and cross-domain features from historical contexts, time series foundation models are designed to accommodate a wider spectrum of prediction scenarios. The concept of unified forecasting is gradually reshaping the conventional practice of task-specific training strategy.

However, existing models face performance bottlenecks in multivariate forecasting. Firstly, as shown in Fig. 1, real-world time series are inherently heterogeneous: (a) within a single sequence, and across different variables from (b) the same or (c) different scenarios. The complex temporal patterns and shifting data distributions pose significant challenges for a single model to extract high-quality features. Meanwhile, recent research has mainly focused on building encoder-only architectures to model temporal dependencies across tokens.

S. Xiao and K. Li are with the China Telecom Research Institute, Beijing 102209, China. E-mail: shixh6@chinatelecom.cn; lik24@chinatelecom.cn.

A. Liang and Y. Sun are with the School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: liangaobo@bupt.edu.cn; sunyan@bupt.edu.cn.

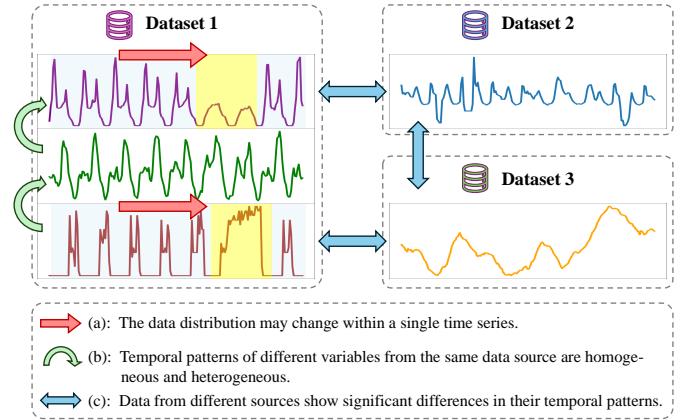


Fig. 1. Homogeneous or heterogeneous time series may exhibit differences in data distribution across (a) different time intervals within a single sequence, (b) varying sequences from the same data source and (c) time series from different data sources.

Although these models often achieve higher accuracy on specific datasets, they tend to suffer from limited generalization. In contrast, generative decoder-only architectures offer better scalability and are becoming an emerging focus of research. Moreover, multivariate time series usually exhibit complex inter-series dependencies, including immediate or delayed correlations in trends and seasonality. Most previous works adopt channel-independent modeling for both pretraining and finetuning stages, where sequences from different variables are processed independently. Considering the significant variation in the number of variables across different real-world applications, it is impractical to rely on a single model structure for all possible multivariate forecasting tasks. Nevertheless, it is undeniable that the channel-independent strategy overlooks valuable inter-variable dependencies, which may become a bottleneck limiting the model's adaptability on specific datasets. Some recent works, such as Moirai(Woo et al., 2024) and Timer-XL(Liu et al., 2024c), have established a unified training paradigm to support multivariate pretraining and finetuning. However, these models try to establish relations among tokens from all input variables without any filtering mechanisms. With the assumption that all input sequences are interrelated, the model may face increased learning cost and introduce noise to tokens from weakly correlated or unrelated variables.

To address the aforementioned issues, we propose **Time Tracker**, a novel generative time series forecasting model architecture. To tackle the problem of highly heterogeneous distributions of time series, we are inspired by sparse mixture of experts (MoE) to assign time series with diverse

distributions to refined experts. Such design alleviates the learning difficulty of a single neural network in capturing specific temporal patterns while reducing the computational cost during inference. In addition, we propose Any-variate Attention (AVA) to enable a unified model structure to seamlessly handle both univariate and multivariate time series causally. Typically, a well pretrained time series foundation model should be equipped with generalizable parameters by training it on data with diverse distributions, achieving robust performance across varied tasks and domains. Considering that the variable numbers and the relations among them may vary significantly across application scenarios, it is unrealistic to generalize the inter-series dependencies of different domains. Therefore, we adopt the channel-independent strategy during pretraining to ensure the model's generalization capacity.

To further adapt the pretrained model to unseen datasets, we perform finetuning stage which in contrast transfers AVA to capturing data-specific inter-series dependencies. We first design a frequency-based graph learning layer to compute the relationship probability between the variables, then use the reparameterization technique to generate the adjacency matrix. We inject such relational information into AVA through the Kronecker product of the adjacency matrix and the causal attention time mask matrix. Consequently, the model can precisely control the interactions of tokens between different variables in a generative manner. In summary, our main contributions are as follows:

- 1) We propose to assign sequence tokens belonging to different data distributions to specific expert networks to reduce the learning difficulty of specific data distributions and improve prediction performance.
- 2) We propose to combine the context-aware graph neural network with causal attention to better adapt to the downstream tasks of multivariate metric data. While the model works in a generative manner within different variables, it captures multivariable dependencies more accurately.
- 3) We pretrain our model on the Unified Time Series Dataset (UTSD) and conduct experiments on multivariate forecasting, zero-shot learning and few-shot learning. Timer Tracker achieves SOTA performance compared to other benchmark models.

II. RELATED WORK

A. Large Time Series Models

Recent advances in pre-training on large-scale sequence data have significantly benefited modality understanding in natural language processing(Grattafiori et al., 2024) and computer vision(Liu et al., 2021; Kirillov et al., 2023). The similar trends has been extended to time series modeling. However, most methods(Wu et al., 2021; Zeng et al., 2023; Nie et al., 2023) are limited in model scale or in-domain applicability, which results in weak generalization ability. When encountering temporal patterns or data distributions previously unseen, these models often require re-training or extensive fine-tuning, significantly increasing deployment cost and reducing practical scalability. To address this limitation, LTSM(Liu et al.,

2024b) are being explored through large-scale pre-training to enhance zero-shot generalization across diverse scenarios. A key challenge lies in managing the inherent heterogeneity of time series data, including variations in domain, frequency and semantics. Some approaches introduce tokenization re-programming framework(Chen, 2024) and attempt to leverage the generalization capability of existing LLMs to adapt to the data distribution of time series. Time-LLM(Jin et al., 2024) generates prompt embeddings by manually crafting dataset-specific prompts and proposes a reprogramming method to map time series into text prototypes, enabling direct prediction using existing LLMs. Lag-Llama(Rasul et al., 2023) maps time series into text prototypes by combining sequence samples at specific time lags with timestamps at multiple time intervals. Chronos(Ansari et al., 2024) tokenizes time series into discrete bins through simple scaling and quantization of real values, achieving time series forecasting through minimizing the loss of a classification task. However, these methods are either reliant on the quality of training datasets or require manual specification of prompts or data sampling rules, which results in higher training costs. Alternatively, recent studies focus on model architectures designed specifically for time series forecasting. MOMENT(Goswami et al., 2024) follows the paradigm of pre-trained NLP models such as BERT(Devlin et al., 2019) by employing a reconstruction-based objective to train a Transformer Encoder as a feature extractor, with task-specific output heads designed for different downstream tasks. Although MOMENT is a general-purpose model tailored for time series analysis, it still requires fine-tuning for specific downstream tasks and cannot be directly applied to forecasting tasks. TimesFM(Das et al., 2024) employs randomly sized masks to enable training with variable-length inputs to fixed-length outputs. Timer(Liu et al., 2024b) performs next-token prediction and uses the causal attention mechanism to model time series in an autoregressive manner. Nevertheless, these models process time series in a channel-independent way and overlook the inter-series dependencies among different variables in both pre-training and fine-tuning stages. To address this issue, Moirai(Woo et al., 2024) proposes a unified training strategy that allows for multivariate time series predictions. However, Moirai implicitly assumes that the input variables from the same data source are mutually correlated, which may introduce noise to each token due to capturing the dependencies of heterogeneous sequences.

B. Mixture of Experts

Mixture of Experts (MoE) has emerged as a solution to scale model capacity efficiently without proportionally increasing computational costs. Through dynamically routing inputs to specialized subnetworks, MOE enables the construction of large-scale models with sparse and resource-efficient computation. Recent works like Time-MOE(Shi et al., 2024) and Moirai-MOE(Liu et al., 2024d) assign different tokens to distinct experts to enhance the scalability and convergency of the original models. However, the in-depth motivation of the token-wise routing approach is ambiguous and ignore the statistical information of the original time series. In addition,

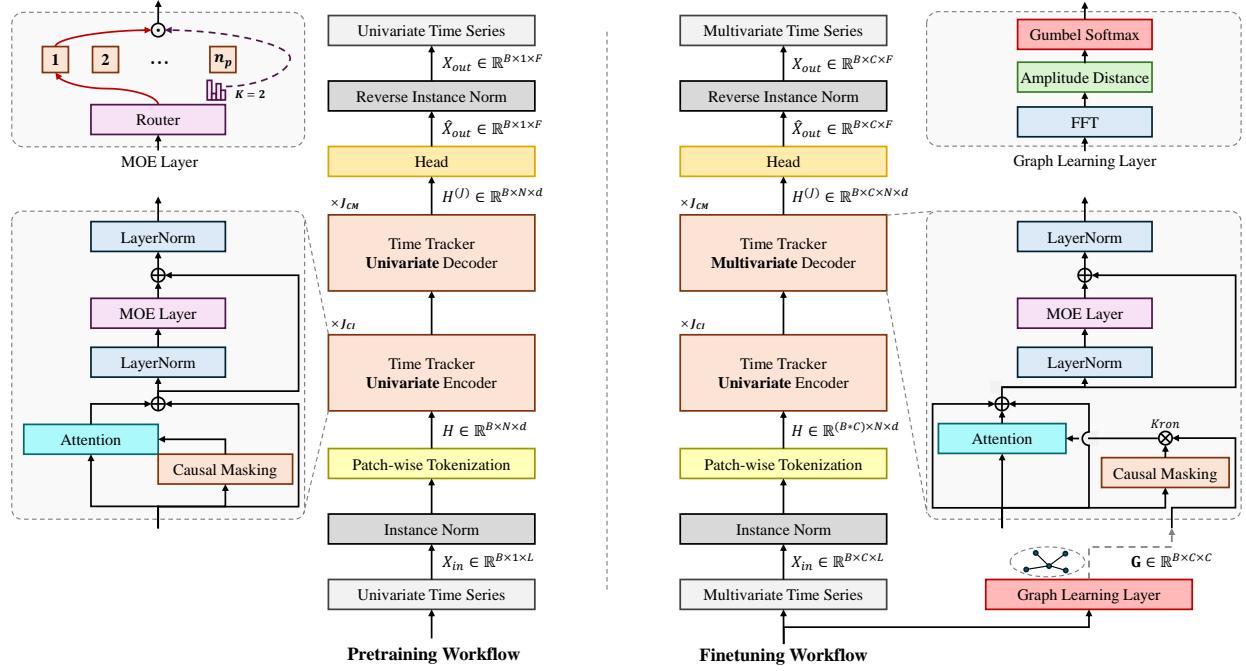


Fig. 2. The model architecture of Time Tracker. The workflow of pretraining stage and finetune stage is on the left and right side respectively.

existing works follow the auxiliary-loss-based load-balance strategy in Switch transformers(Fedus et al., 2022) which may lead to suboptimal performance due to imbalanced loss weight assignment and potential difficulties in achieving equitable expert utilization(Dai et al., 2024).

C. Graph Neural Networks

GNN has been widely used for spatial temporal modeling in multivariate forecasting. Previous works such as STGCN and Graph Wavenet use GNN to model sensors of different roads as nodes in the graph structure. Although the predicting performance is improved, the model needs predefined graph structure, which is usually not achievable in more complex scenarios. MTGNN introduces an adaptive graph learning layer to automatically generate an optimal adjacency matrix tailored to specific data. However, when faced with new scenarios involving different nodes, the model requires retraining, which limits its scalability. To address this issue, STEP proposes an instance-wise graph learning module to adapt to downstream tasks with any possible variables. STEP combines Bernoulli sampling with gumbel softmax so that the model can generate adjacency matrix based on the similarity between embeddings of different time series.

III. METHODOLOGY

In this section, we propose Time Tracker, a fundamental time series model built for multivariate forecasting. As shown in Fig 2, Time Tracker is based on a decoder-only Transformer architecture. We pretrain the model in the channel-independent manner where each univariate time series is fed into the model separately. In this setup, all model parameters are

shared across different time series to capture diverse temporal evaluation patterns, therefore enhancing its generalization ability to unseen or heterogeneous time series data. To achieve more accurate prediction, we argue to capture inter-series dependencies while applying the model on specific datasets. In the fine-tuning stage, we design an adaptive graph learning layer to generate instance-wise adjacency matrices. We further combine the causal mask with the adjacency matrix to capture inter-series dependencies while keeping the autoregressive predicting scheme.

A. Patch-wise Next Token Prediction

Considering the input multivariate time series data $X_{in} = \{x_{1,t}, x_{2,t}, \dots, x_{C,t}\}_{t=1}^L$, our objective is to predict the future values $X_{out} = \{x_{1,t}, x_{2,t}, \dots, x_{C,t}\}_{t=L+1}^{L+F}$, where C denotes the variable dimension, L is the length of historical look-back window and F represents the forecasting horizons. We first segment each series into $X_p = \{p_{1,\tau}, p_{2,\tau}, \dots, p_{C,\tau}\}_{\tau=1}^N$ where each patch $p_{i,j} = x_{i,(j-1)P+1:jP}$ corresponds to P consecutive time points from the input series. The patches are extracted using a sliding window with stride S , producing $N = \lceil \frac{L-P}{S} + 1 \rceil$ patch-wise tokens. We set $F = P$ so that the model learns to predict the next patch of length P given the previous $N - 1$ patches. We pass each token in X_p through a linear projector to form multivariate patch-wise tokens $H = \{h_{1,\tau}, h_{2,\tau}, \dots, h_{C,\tau}\}_{\tau=1}^N$ for further processing, where $h_{i,j} \in \mathbb{R}^d$ and d is the token dimension.

B. Frequency-based Adaptive Graph Learning Layer

To capture the underlying relationships among time series, we explore the similarity between different series in the frequency domain. Unlike direct modeling in the time

domain, this approach allows us to extract invariant spectral features that are more robust to localized variations and more intuitive in terms of representing periodic patterns. Therefore, it facilitates the learning of more stable and generalizable inter-series relationships. We use real fast Fourier transform(RFFT) to represent X_{in} in the frequency domain, denoted as $X_{in}^f \in \mathbb{R}^{C \times (1 + \frac{L}{2})}$. Subsequently, we compute the differences in amplitude values across various frequencies to derive a probability matrix $\tilde{\mathcal{G}}$ that quantifies the inter-variable correlations, as shown in eq 1:

$$\begin{aligned} X_{in}^f &= \text{RFFT}(X_{in}) = \{x_{1,t}^f, x_{2,t}^f, \dots, x_{C,t}^f\}_{t=1}^{1+\frac{L}{2}} \\ \mathbf{D}_{i,j} &= \sum_{t=1}^{1+\frac{L}{2}} |x_{i,t}^f - x_{j,t}^f|, \text{ for } i, j \in \{1, 2, \dots, C\} \\ \tilde{\mathbf{G}}_{i,j} &= \begin{cases} \frac{\alpha \mathbf{D}_{i,j}}{\text{argmax}(\mathbf{D})} & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}, \end{aligned} \quad (1)$$

where $\alpha \in (0, 1)$ is the weighting factor to avoid absolute relationship between different variables. Given the probability matrix, we sample a binary adjacency matrix A using the Gumbel-Softmax with reparameterization trick to enable differentiable Bernoulli sampling. Specifically, we introduce a random value $\epsilon \in \mathbb{R}^2$ sampled from Gumbel(0, 1) distribution and compute \mathbf{G} through eq 2, where $\tau \rightarrow 0$ is the temperature parameter to achieve approximate Bernoulli sampling.

$$\mathbf{G}_{i,j} = \text{Softmax}((\tilde{\mathbf{G}}_{i,j} + \epsilon)/\tau) \quad (2)$$

C. Any-variate Causal Attention

The basic causal attention mechanism constrains each token to focus only on preceding tokens through a causal mask. Prior works typically apply this approach to single-channel data or channel-independent models, overlooking inter-series dependencies. To address this, we aim to develop an any-variate causal attention mechanism that incorporates cross-variable information while preserving temporal causality. Following (Woo et al., 2024), we flatten the multivariate tokens H to $\mathbb{R}^{(C*N) \times d}$ and calculate the attention scores as defined in eq 3:

$$\begin{aligned} \tilde{\mathcal{A}}_{(i-1)N+m, (j-1)N+n} &= (\mathbf{W}_Q h_{i,m})^\top \mathbf{R}_\Theta m-n (\mathbf{W}_K h_{j,n}) \\ &\quad + u \cdot \mathbf{1}(i = j) + v \cdot \mathbf{1}(i \neq j), \\ i, j &\in \{1, 2, \dots, C\}, m, n \in \{1, 2, \dots, N\}, \end{aligned} \quad (3)$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d \times d}$ are the weight matrices, \mathbf{R}_Θ refers to the rotary matrix(Su et al., 2024), $\mathbf{1}$ is the indicator function and u, v are two learnable parameters that ensure the permutation equivalence of \mathcal{A} across variables. We further use a causal mask \mathbf{M} defined in eq 6 to properly construct the causal relation among tokens within different variables. We first calculate the Kronecker product of the adjacency matrix $\mathbf{G} \in \mathbb{R}^{C \times C}$ and the temporal causal mask $\mathbf{T} \in \mathbb{R}^{N \times N}$ and save the results in $\tilde{\mathbf{M}}$, as defined in eq 4. \mathbf{T} is a lower triangular mask that masks the attention scores from each token to future positions. The temporal constraint is then broadcast to related variables via the adjacency matrix \mathbf{G} .

Specifically, for any $h_{i,m}$ and $h_{j,n}$ with $j \neq i$, if $\mathbf{G}_{i,j} = 1$, then $h_{i,m}$ and $h_{j,n}$ are mutually dependent for all $n \leq m$.

$$\begin{aligned} \tilde{\mathbf{M}}_{(i-1)N+m, (j-1)N+n} &= \mathbf{G}_{i,j} \mathbf{T}_{m,n}, \\ \mathbf{T}_{m,n} &= \begin{cases} 0 & \text{if } m \leq n \\ 1 & \text{otherwise}, \end{cases} \\ i, j &\in \{1, 2, \dots, C\}, m, n \in \{1, 2, \dots, N\}, \end{aligned} \quad (4)$$

Subsequently, we construct the mask matrix \mathbf{M} through eq 5. Finally, we apply \mathbf{M} to the attention scores \mathcal{A} to establish causal and variable-wise dependencies among the multivariate tokens, as defined in eq 6:

$$\begin{aligned} \mathbf{M}_{i,j} &= \text{mask}(\tilde{\mathbf{M}}_{i,j}), i, j \in \{1, 2, \dots, C * N\}, \\ \text{mask}(x) &= \begin{cases} -\infty & \text{if } x = 0 \\ 1 & \text{otherwise}, \end{cases} \end{aligned} \quad (5)$$

$$\begin{aligned} \text{Attention}(H^{(\ell)}) &= \text{Unflat}\left(\mathcal{S} \cdot \text{Flat}(W_V H^{(\ell)})\right) \\ \mathcal{S}^{(\ell)} &= \text{Softmax}\left(\frac{\mathcal{A} + \mathbf{M}}{\sqrt{d}}\right) \end{aligned} \quad (6)$$

where $W_V \in \mathbb{R}^{d \times d}$ is the weight matrix for the values vectors, $\mathcal{S}^{(\ell)}$ is the attention scores, $\ell \in \{1, 2, \dots, J\}$ and J is the number of model layers. Note that $H^{(1)}$ refers to the results of patch-wise tokens, Flat and Unflat are the functions that convert the shape of inputs to $\mathbb{R}^{(C*N) \times d}$ and $\mathbb{R}^{C \times N \times d}$ respectively.

D. Channel-wise Mixture of Experts

Real-world time series often contain diverse and non-stationary patterns across different variables, making it difficult for a single model to generalize well across all conditions(Sun et al., 2024). In addition, time series data inherently exhibit strict temporal dependencies, where adjacent tokens are often semantically correlated. This motivate us to assign expert networks to tokens within the same univariate sequence to enhance the model's stability and expressiveness. Specifically, we replace the each feedforward network (FFN)(Vaswani et al., 2017) with an MOE layer, containing n_s shared experts and n_p private experts. Each expert network keeps the same architecture of a standard FFN. We design a token cluster $\mathcal{G} \in \mathbb{R}^{n_p \times d}$ to match the tokens within a univariate sequence to the private expert responsible for the corresponding data distribution. To avoid the issue of routing collapse(Shazeer et al., 2017), we introduce a channel-wise bias factor $b \in \mathbb{R}^{n_p}$ to achieve auxiliary-loss-free load balance(Wang et al., 2024). The entire computation process of the attention and MOE layer in algorithm 1, where the function $\text{Attn}(\cdot)$ corresponds to eq 6, K refers to the top-k elements and S_FFN, P_FFN are the shared and private experts.

E. Pre-training Workflow

Pretrained models are designed to capture generic and transferable patterns from large-scale data, enabling better generalization across different tasks and domains(Devlin et al.,

Algorithm 1 The implementation of the Encoder

Input: Input sequence tokens $H^{(\ell)} \in \mathbb{R}^{C \times N \times d}$

Output: Output representations $H^{(\ell+1)} \in \mathbb{R}^{C \times N \times d}$

- 1: $\hat{H}^{(\ell)} \in \mathbb{R}^{C \times N \times d} \leftarrow \text{RMSNorm}(\text{Attn}(H^{(\ell)}) + H^{(\ell)})$
- 2: $s \in \mathbb{R}^{C \times N \times n_p} \leftarrow \hat{H}^{(\ell)}(\mathcal{G}^{(\ell)})^\top$
- 3: $\bar{s} \in \mathbb{R}^{C \times n_p} \leftarrow \text{Softmax}\left(\frac{1}{N} \sum_{i=1}^N s_{:,i,:}\right)$
- 4: Initialize empty vectors $g \in \mathbb{R}^{C \times N \times n_p}, c \in \mathbb{R}^{n_p}$
- 5: **for** $i = 1$ to C **do**
- 6: **for** $j = 1$ to n_p **do**
- 7: **if** $\bar{s}_{i,j} + b_j^{(\ell)} \in \text{TopK}(\{\bar{s}_{i,k} + b_k^{(\ell)} | 1 \leq k \leq n_p\}, K)$ **then**
- 8: $g_{i,:,j} \leftarrow \bar{s}_{i,j}$
- 9: $c_j \leftarrow c_j + 1$
- 10: **else**
- 11: $g_{i,:,j} \leftarrow 0$
- 12: **end if**
- 13: **end for**
- 14: **end for**
- 15: $\bar{c} \leftarrow \frac{1}{n_p} \sum_{i=1}^{n_p} c_i$
- 16: **for** $i = 1$ to n_p **do**
- 17: $e = \bar{c} - c_i$
- 18: $b_i^{(\ell+1)} = b_i^{(\ell)} + u * \text{sign}(e)$
- 19: **end for**
- 20: $\tilde{H}_s^{(\ell+1)} \in \mathbb{R}^{C \times N \times d} \leftarrow \frac{1}{n_s} \sum_{i=1}^{n_s} S_FFN_i(\hat{H}^{(\ell)})$
- 21: $\tilde{H}_p^{(\ell+1)} \in \mathbb{R}^{C \times N \times d} \leftarrow \sum_{i=1}^{n_p} P_FFN_i(\hat{H}^{(\ell)}) * g_{:,i,:}$
- 22: $\tilde{H}^{(\ell+1)} \in \mathbb{R}^{C \times N \times d} \leftarrow \tilde{H}_s^{(\ell+1)} + \tilde{H}_p^{(\ell+1)}$
- 23: $H^{(\ell+1)} \in \mathbb{R}^{C \times N \times d} \leftarrow \text{RMSNorm}(\tilde{H}^{(\ell+1)} + \hat{H}^{(\ell)})$
- 24: **return** $H^{(\ell+1)}$

2019). In the context of time series forecasting, processing each series individually encourages the model to learn diverse temporal patterns from individual series, thereby enhancing the model's generalization capability(Nie et al., 2023). Besides, multivariate time series from different scenarios often exhibit significant discrepancies in both dimensionality and the underlying inter-series relationships. Modeling such heterogeneous dependencies in a unified pretraining framework poses challenges in terms of scalability and representation consistency. Therefore, we adopt a channel-independent data loading strategy in the pretraining stage. Given a training dataset with $num_{\mathcal{D}}$ multivariate subsets, we extract univariate sequences from each variable to construct the training samples. Specifically, for a multivariate subset \mathcal{D}_i with data in the shape of $\mathbb{R}^{C_i \times T}$, where T is the length of each time series and $i \in \{1, 2, \dots, num_{\mathcal{D}}\}$, we generate training instances of the shape $1 \times (L + F)$. Accordingly, the total number of training samples extracted from \mathcal{D}_i is $C_i \times (T - L - F)$.

F. Fine-tuning Workflow

The primary objective in the fine-tuning stage is to adapt the pretrained model to the target dataset, allowing it to capture

task-specific patterns that may not have been learned during pretraining. Previous works usually maintain consistent data loading strategies in both pretraining and fine-tuning stages, making it difficult for the model to balance the performance in zero-shot and few-shot tasks. Inspired from iTransformer(Liu et al., 2024a), we argue that capturing inter-series dependencies of multivariate data in specific datasets could provide better adaptability. This motivates us to introduce the graph learning module in the fine-tuning stage, enabling the model to additionally capture inter-series dependencies within multivariate inputs. Different from the pretraining stage, we keep the original shape of multivariate instance as $\mathbb{R}^{C \times (L+F)}$. For a batched input data $X_{in} \in \mathbb{R}^{B \times C \times L}$, we first flatten it into the shape of $\mathbb{R}^{(B*C) \times 1 \times L}$ to maintain univariate processing in the first J_{CI} layers, producing $H^{(J_{CI})} \in \mathbb{R}^{(B*C) \times 1 \times N \times D}$. Consequently, we concatenate the tokens and feed $H^{(\ell_{CM})} \in \mathbb{R}^{B \times (C*N) \times D}$ into the last J_{CM} layers. Through the adjacency matrix and any-variate causal attention layers in SecIII-B and SecIII-C, we capture inter-series dependencies in a causal manner to achieve better adaptation.

IV. EXPERIMENTS**A. Datasets**

We follow the work of Timer and pretrain our model on Unified Time Series Dataset (UTSD)(Liu et al., 2024b). UTSD is a large time series dataset derived from publicly available online data repositories and real-world machine operation data. It covers seven major domains including energy, environment, health, IoT, nature, transportation, and networks, with up to 1 billion time points. Each subset within UTSD is analyzed in terms of stationarity and predictability to ensure an appropriate level of inherent complexity. To evaluate the generality and adaptivity of our model, we also choose six well-known real-world datasets including Weather, Electricity, Traffic and ETT series. The details of these datasets are shown in table I. All these datasets can be reached through previous works(Wu et al., 2021; Liu et al., 2024a).

TABLE I
STATISTICS OF ALL DATASETS.

Dataset	Variables	Frequency	Length
Weather	21	10 min	52696
Electricity	321	1 hour	26304
Traffic	862	1 hour	17544
ETTh1	7	1 hour	17420
ETTh2	7	1 hour	17420
ETM1	7	15 min	69680
ETM2	7	15 min	69680

B. Multivariate Forecasting

1) **Setup:** To evaluate the effectiveness of the proposed model structure, we follow iTransformer(Liu et al., 2024a) and conduct experiments of multivariate forecasting on the six benchmark datasets. The lengths of training, validation and test sets are recorded in table I. We set $L = 96$ and $F = 96$ and present the results of MSE and MAE in table II. Since we aim to directly predict multivariate series, we activate

TABLE II

EXPERIMENTAL RESULTS OF LONG-TERM MULTIVARIATE TIME SERIES FORECASTING TASK ON 7 REAL-WORLD DATASETS. THE BEST RESULTS ARE IN **BOLD**. THE LENGTHS OF INPUT AND PREDICTING SERIES ARE $L = 96$ AND $F = 96$ RESPECTIVELY. THE MODELING TYPES ARE ANNOTATED BEHIND EACH MODEL, WHERE **CI** REFERS TO CHANNEL-INDEPENDENT AND **CM** REFERS TO CHANNEL-MIXING. THESE ANNOTATIONS INDICATE WHETHER THE MODEL CONSIDERS INTER-SERIES DEPENDENCIES.

Models	Time Tracker (CM)		Timer (CI)		iTTransformer (CM)		PatchTST (CI)		Autoformer (CM)		DLinear (CI)		TimesNet (CI)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	0.169	0.213	0.176	0.217	0.175	0.216	0.172	0.214	0.249	0.329	0.198	0.260	0.172	0.220
Traffic	0.370	0.249	0.407	0.260	0.392	0.263	0.444	0.283	0.597	0.371	0.652	0.386	0.593	0.321
Electricity	0.141	0.233	0.158	0.242	0.153	0.245	0.167	0.253	0.196	0.313	0.195	0.278	0.168	0.272
ETTh1	0.380	0.400	0.379	0.399	0.394	0.410	0.380	0.399	0.435	0.446	0.391	0.403	0.384	0.402
ETTh2	0.294	0.347	0.309	0.356	0.303	0.353	0.293	0.342	0.332	0.368	0.375	0.397	0.340	0.374
ETTm1	0.325	0.363	0.330	0.367	0.339	0.374	0.326	0.363	0.510	0.492	0.344	0.372	0.338	0.375
ETTm2	0.178	0.261	0.180	0.201	0.189	0.274	0.177	0.261	0.205	0.293	0.190	0.287	0.187	0.267

TABLE III

ZERO-SHOT RESULTS OF LONG-TERM MULTIVARIATE TIME SERIES FORECASTING TASK ON 7 REAL-WORLD DATASETS. THE BEST RESULTS ARE IN **BOLD**. TIME TRACKER IS PRETRAINED ON UTSD DATASET, WHICH IS KEPT SAME FOR PRETRAINING TIME-XL AND TIMER. THE LENGTHS OF INPUT AND PREDICTING SERIES ARE $L = 672$ AND $F = 96$ RESPECTIVELY.

Models	Time Tracker _{Base} (Ours)		Timer _{Base} (2024b)		Moirai _{Base} (2024)		MOMENT (2024)		Chronos _{Base} (2024)	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	0.173	0.221	0.172	0.224	0.175	0.210	0.243	0.255	0.203	0.238
Traffic	0.413	0.286	0.471	0.322	0.449	0.306	0.507	0.331	0.440	0.299
Electricity	0.150	0.241	0.165	0.257	0.201	0.278	0.291	0.355	0.153	0.231
ETTh1	0.376	0.400	0.388	0.406	0.392	0.402	0.688	0.557	0.440	0.393
ETTh2	0.303	0.354	0.305	0.355	0.284	0.331	0.342	0.396	0.308	0.343
ETTm1	0.386	0.396	0.552	0.473	0.447	0.403	0.654	0.527	0.454	0.408
ETTm2	0.189	0.273	0.222	0.294	0.219	0.290	0.260	0.335	0.199	0.274

TABLE IV

FEW-SHOT RESULTS OF LONG-TERM MULTIVARIATE TIME SERIES FORECASTING TASK ON 7 REAL-WORLD DATASETS. THE BEST RESULTS ARE IN **BOLD**. THE MODELS ARE ALL PRETRAINED ON UTSD DATASET, WHILE TIME TRACKER ADOPTS MULTIVARIATE FINETUNING AND TIMER USES UNIVARIATE FINETUNING. THE LENGTHS OF INPUT AND PREDICTING SERIES ARE $L = 672$ AND $F = 96$ RESPECTIVELY.

Datasets	Weather		Traffic		Electricity		ETTh1		ETTh2		ETTm1		ETTm2	
	MSE	MAE												
Time Tracker	0.154	0.200	0.353	0.244	0.128	0.221	0.364	0.391	0.294	0.351	0.341	0.369	0.171	0.254
Timer	0.151	0.198	0.362	0.247	0.132	0.225	0.378	0.398	0.296	0.356	0.395	0.393	0.177	0.256

the graph learning layer and use the adjacency matrix \mathbf{G} in all J layers. For Weather, Traffic and Electricity that have more variables and exhibit better seasonality, we set $J = 4$ to capture more complex inter-series dependencies. For ETT datasets with relatively weak relationships between variables, we set $J = 2$ to slightly weaken the emphasis on inter-series dependencies.

2) *Results:* Generally, Time Tracker achieves advanced predicting performance on all the datasets among both channel-mixing models and channel independent models. Compared to iTTransformer, the SOTA channel-mixing model, Time Tracker

reduces the MSE and MAE by 4.42% and 3.23% on average. When dealing with datasets with weak inter-series relationships, Time Tracker can also reach or outperform the SOTA channel-independent models, including a LTSV benchmark model Timer. This indicates that the structure of Time Tracker can effectively capture both temporal dependencies within the sequence and inter-series dependencies, regardless of whether the data exhibits high or low variable correlations.

C. Zero-shot Learning

1) *Setup:* In this section, we pretrain Time Tracker in the univariate manner on UTSD and conduct experiments on six well-known datasets. For all the baseline models, we set the input size as $L = 672$ and patch size as $P = 96$. We record the MSE and MAE results of $F = 96$ in table III. Note that for fair comparison, we follow the model structure of other baseline models and pretrain them with $L = 672$. We use the pretrained version of baseline models to get zero-shot results.

2) *Results:* Time Tracker achieves superior predicting performance. Compared to the SOTA multivariate pretrained model Moirai, the MSE and MAE are reduced by 9.59% and 2.79%. This improvement comes to 10.98% and 6.77% relative to the results obtained by Timer, which is pretrained in a univariate manner. Generally, pretraining models in a univariate way could help enhancing the model's generality. What's more, we believe that the MOE layers can better generalize across diverse temporal patterns by assigning specific experts to different types of time series. In the absence of MoE, FFN layers have to fit all data distributions simultaneously, which may results in suboptimal network parameters when dealing with newly seen data. By contrast, MoE enables experts to specialize in prototype-like data distribution. Each expert adapts to a specific distribution in the high-dimensional space. When dealing with data within new distributions, MOE enables the model to activate parameters closer to the target pattern rather than relying on the averaged ones across all pretrained data.

D. Few-shot Learning

1) *Setup:* In this section, to evaluate the model's adaptability, we conduct few-shot learning experiments on six datasets. Different from the pretraining stage, we introduce the graph learning layer to generate an adjacency matrix \mathbf{G} to guide the interaction among tokens from different variables. We set the first $\ell_u = 7$ Time Tracker layers remaining the parameters to process the input data in a channel-independent way. The last $\ell_m = 1$ Time Tracker layer will concatenate all sequences to the multivariate form and capture inter-series dependencies. Since the parameters of the attention layers and normalization layers are independent of the number of tokens, all parameters from the pretrained model will be kept to enable faster convergence. We use Timer as the benchmark model and conduct univariate finetuning. All the time series will be input independently to the model. We use only 20% of the dataset as the training set while 20% as the testing set and record the MSE and MAE results in table IV.

2) *Results:* Compared to the univariately finetuned model Timer, Time Tracker demonstrates superior adaptability to complex datasets, particularly those involving multiple variables with stronger interdependencies. The average improvements of MSE and MAE are 3.56% and 2.34%. Our proposed graph learning module adaptively constructs inter-series relationships for previously unseen multivariate data and integrates the relational information into the any-variate attention layer. Compared to maintaining a univariate finetuning strategy, our design facilitates propagating the temporal patterns across

sequence tokens from variables with strong correlations. With the help of inter-series dependencies, Time Tracker exhibits better adaptability to specific datasets.

V. CONCLUSION

In this paper, we propose Time Tracker, a large time series model architecture for multivariate time series forecasting. We seamlessly integrate MOE with Transformer to assign different expert networks to sequence tokens from time series with varying data distributions, aiming to produce higher-quality features. We design an any-variate causal attention mechanism that can process tokens from any number of variables with the same model structure. Such a module allows for different data loading strategies in the pretraining and finetuning stages. We pretrain the model under a univariate setting and seamlessly transfer the obtained parameters to the multivariate finetuning stage. The decoupled design effectively enhances the model's generalization to unseen data distributions and adaptability to specific datasets. Future research will be done to explore more efficient model architectures.

REFERENCES

- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.
- Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.
- Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International*

- Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023.* OpenReview.net, 2023.
- Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a.
- Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32369–32399, 2024b.
- Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024c.
- Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22584–22591, 2024.
- Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-lm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.
- Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riahi, Nadhir Hassen, et al. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.
- Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.
- Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: a family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 16115–16152, 2024.
- Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.
- Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.
- Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024d.
- Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts, 2024. URL <https://arxiv.org/abs/2409.16040>.
- Damai Dai, Chengqi Deng, Chenggang Zhao, Rx Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297, 2024.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- Yanru Sun, Zongxia Xie, Emadeldeen Eldele, Dongyue Chen, Qinghua Hu, and Min Wu. Learning pattern-specific experts for time series forecasting under patch-level distribution shift. *arXiv preprint arXiv:2410.09836*, 2024.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.