AdaMixT: Adaptive Weighted Mixture of Multi-Scale Expert Transformers for Time Series Forecasting

Huanyao Zhang[1,2,*] , **Jiaye Lin**[3,*] , **Wentao Zhang**[2] , **Haitao Yuan**[3,†] and **Guoliang Li**[3,†]

[1]School of Computer Science, Peking University, Beijing, China
[2]Center for Machine Learning Research, Peking University, Beijing, China
[3]Tsinghua University, Beijing, China

Abstract

Multivariate time series forecasting involves predicting future values based on historical observations. However, existing approaches primarily rely on predefined single-scale patches or lack effective mechanisms for multi-scale feature fusion. These limitations hinder them from fully capturing the complex patterns inherent in time series, leading to constrained performance and insufficient generalizability. To address these challenges, we propose a novel architecture named Adaptive Weighted Mixture of Multi-Scale Expert Transformers (AdaMixT). Specifically, AdaMixT introduces various patches and leverages both General Pre-trained Models (GPM) and Domain-specific Models (DSM) for multi-scale feature extraction. To accommodate the heterogeneity of temporal features, AdaMixT incorporates a gating network that dynamically allocates weights among different experts, enabling more accurate predictions through adaptive multi-scale fusion. Comprehensive experiments on eight widely used benchmarks, including Weather, Traffic, Electricity, ILI, and four ETT datasets, consistently demonstrate the effectiveness of AdaMixT in real-world scenarios.

1 Introduction

Time series forecasting is essential for various fields, aiming to accurately present future values according to historical observations. The rapid advancement of deep learning has spurred significant research in this area, with applications in traffic prediction [Zhao *et al.*, 2024; Jiang *et al.*, 2024; Yuan and Li, 2021; Chen *et al.*, 2024], recommender systems [Lin *et al.*, 2024a; Lin *et al.*, 2024b], and weather forecasting [Zhou *et al.*, 2021; Liu *et al.*, 2025; Miao *et al.*, 2024b].

The recent success of attention mechanisms has prompted researchers to investigate the potential of Transformer-based models by redefining time series forecasting tasks as the future token prediction [He *et al.*, 2024b; Chen *et al.*, 2025; Yuan *et al.*, 2024]. Existing research in this field can be categorized into two classes based on different tokenization methods, as illustrated in Figure 1. The first approach uses timestamps as tokens, exemplified by models such as Autoformer [Wu *et al.*, 2021] and FEDformer [Zhou *et al.*, 2022]. The second approach adopts patch-level tokenization, where a patch represents a window of timestamps, with PatchTST [Nie *et al.*, 2022] being a notable example. Compared with timestamp-level tokenization, patch-level tokenization more effectively captures temporal patterns, thereby achieving superior forecasting performance. Despite their remarkable success, these methods still face challenges related to limited generalizability. Most existing research focuses primarily on single-scale temporal features, which lack a careful consideration of multi-scale characteristics in time series data. Furthermore, the effective fusion of multi-scale features has not been thoroughly explored, leading to limitations in forecasting under complex scenarios.

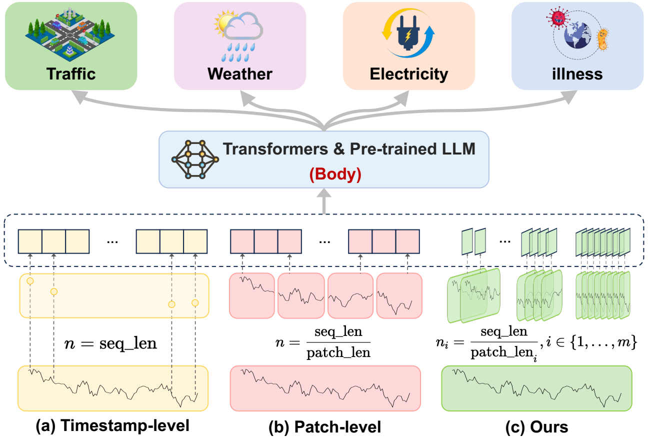

Traffic
Weather
Electricity
illness
Transformers & Pre-trained LLM
(Body)
$n = \text{seq\_len}$
$n = \frac{\text{seq\_len}}{\text{patch\_len}}$
$n_i = \frac{\text{seq\_len}}{\text{patch\_len}_i}, i \in \{1, \ldots, m\}$
(a) Timestamp-level
(b) Patch-level
(c) Ours

Figure 1: **The existing technologies and our main idea.** (a) Timestamp-level tokenization, where each timestamp is treated as an individual token. (b) Patch-level tokenization, where each time window serves as a token, with $patch\_len$ denoting the length of a patch. (c) Multi-scale feature extraction (Ours), which utilizes m different $patch\_len$ to capture features at varying scales from the time series. In this context, n denotes the number of tokens, and seq_len refers to the length of time series data.

To address the above challenges, we propose a novel time series forecasting architecture named Adaptive Weighted Mixture of Multi-Scale Expert Transformers (AdaMixT). By

[†]Corresponding author.

incorporating multi-scale temporal features, AdaMixT significantly enhances both predictive accuracy and generalizability, providing an innovative approach to multi-scale feature fusion. Specifically, AdaMixT leverages both General Pretrained Models (GPM) and Domain-specific Models (DSM) for time series forecasting, combining the extensive knowledge from GPM with the specialized feature extraction capabilities of DSM to better support downstream tasks. In this framework, input data is segmented into multi-scale patches; smaller patches capture high-frequency features for high-resolution representations, while larger patches capture low-frequency features for low-resolution representations. To effectively integrate multi-scale features, AdaMixT adopts a weighted fusion strategy to assign different weights for the output of each model, generating the final prediction. In summary, we make the main contributions as follows:

- We propose a novel multi-scale patch design that combines GPM with DSM, enabling the capture of both short-term and long-term temporal patterns in time series. This integration leverages the open-world knowledge of pretrained models and the specialized feature extraction capabilities of domain-specific architectures, thereby improving forecasting accuracy and model generalizability.
- To the best of our knowledge, this work is the first to introduce an adaptive mechanism for multi-scale feature fusion in time series forecasting. By utilizing multiple models as experts and dynamically assigning weights based on the characteristics of time series data, our approach significantly enhances the model's adaptability to various tasks, ensuring robust scalability across different real-world application scenarios.
- We conduct extensive experiments across multiple time series forecasting benchmarks, demonstrating that AdaMixT outperforms existing forecasting approaches.

2 Related Work

2.1 Time Series Forecasting

In recent years, considerable research efforts have been directed toward leveraging Transformer-based models for time series forecasting. Early approaches, such as Informer [Zhou *et al.*, 2021], adopt a sequence-to-sequence framework, treating each timestamp as an individual token. Similarly, Autoformer [Wu *et al.*, 2021] incorporates classical analysis concepts such as decomposition and autocorrelation, while FEDformer [Zhou *et al.*, 2022] utilizes a Fourier-enhanced structure to achieve linear computational complexity. However, [Zeng *et al.*, 2023] highlights the limitations of treating each timestamp as a token, particularly in capturing intricate temporal patterns. To overcome this drawback, models such as Crossformer [Zhang and Yan, 2023] and PatchTST [Nie *et al.*, 2022] are inspired by patch-based visual transformers [Dosovitskiy *et al.*, 2020], representing windows of multiple timesteps as patches and using these as tokens for improved performance. Concurrently, the impressive capabilities of Large Language Models (LLMs) have led to their application in time series forecasting. Models like GPT4TS [Zhou *et al.*, 2023] and TIME-LLM [Jin *et al.*, 2023] have shown promising results, further demonstrating the potential of LLMs in this field.

Despite these advancements, existing methods [Miao *et al.*, 2024a; Yuan *et al.*, 2021; Yuan *et al.*, 2020] still face challenges in effectively capturing and fusing multi-scale features. To overcome these limitations, we propose AdaMixT, which enhances robust multi-scale feature extraction and includes an efficient fusion mechanism. By combining the strengths of GPM and DSM, AdaMixT offers a comprehensive and efficient solution for time series analysis.

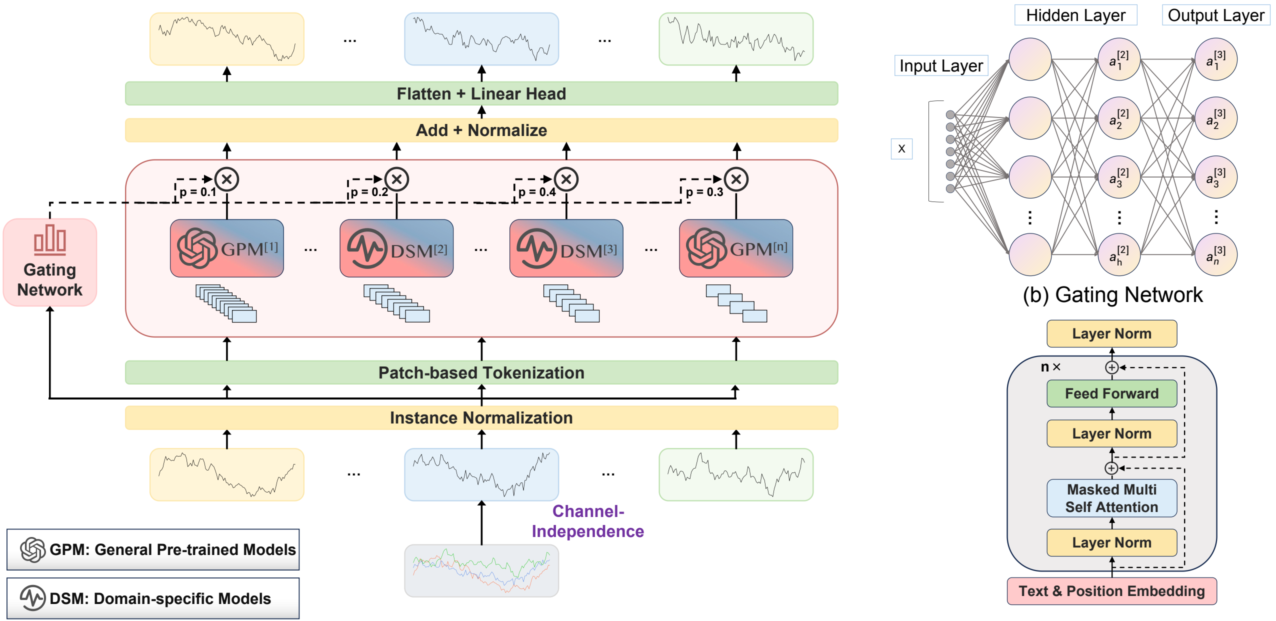
2.2 Multi-scale Feature Learning

The analysis of multi-scale features plays a pivotal role in numerous fields. In computer vision, multi-scale features enable the extraction of information at varying spatial resolutions within an image, facilitating the analysis of both fine-grained local details and overarching global structures [Das and Dutta, 2020; He *et al.*, 2024a]. Meanwhile, multi-scale features have also been widely applied in domains such as knowledge graphs [Li *et al.*, 2025; Li *et al.*, 2022]. In recent years, multi-scale analysis has also been increasingly adopted in time series forecasting. For instance, TimesNet [Wu *et al.*, 2022] transforms one-dimensional sequences into two-dimensional tensors to capture diverse periodic patterns. In addition, MICN [Wang *et al.*, 2023] utilizes convolutions with varying kernel sizes to learn features at multiple temporal scales. TimeMixer [Wang *et al.*, 2024] enhances predictive performance by decomposing multiscale series and effectively blending their seasonal and trend components.

Unlike existing works [Miao *et al.*, 2025; Yuan *et al.*, 2023], AdaMixT learns multi-scale features during the training phase, enabling more efficient feature representation. Furthermore, instead of relying on straightforward feature fusion methods such as concatenation or addition, we design a gating network to score multi-scale temporal features and propose an innovative multi-scale feature fusion mechanism, which further enhances the performance and generalizability.

2.3 Mixture of Experts

Mixture of Experts (MoE) refers to a model consisting of different components, known as experts, each specialized in handling distinct tasks or specific aspects of data. Initially introduced in the literature [Jacobs *et al.*, 1991], it has since been extensively studied and refined in subsequent works [Aljundi *et al.*, 2017]. The advent of sparse-gated MoE [Shazeer *et al.*, 2017], particularly within large Transformer-based language models [Lepikhin *et al.*, 2020], has revitalized this technique, expanding its applicability and effectiveness. Traditional MoEs typically rely on feedforward networks to select models based on specific scenarios. Unlike these designs, AdaMixT introduces multi-scale feature fusion into time series analysis for the first time. By adopting a self-learning mechanism, AdaMixT focuses on feature fusion rather than model selection, enabling the model to automatically identify critical multi-scale features and achieve efficient integration.



(a) AdaMixT Model Overview

(b) Gating Network

(c) GPT-2 (Example of GPM)

Figure 2: **The architecture of AdaMixT.** (a) AdaMixT adopts a channel-independent design, extracting multi-scale features through patches of varying lengths and feeding these features into multiple experts, including GPMs and DSMs. The outputs of experts are dynamically weighted and fused via a gating network, ultimately generating the final prediction. (b) Gating Network dynamically selects and fuses the outputs of multiple experts, where X denotes the time series features before patching, h denotes the number of hidden neurons, and n denotes the number of experts. (c) GPT-2, serving as a representative of the GPM, can efficiently capture general features and provide a powerful foundational representation for subsequent multi-scale feature fusion.

3 Method

In this section, we first provide a formal definition of the problem along with necessary notations and then explain the details of the overall model structure.

3.1 Problem Definition

Multivariate time series forecasting involves predicting future values based on historical observation. The objective is to forecast future values for the next K timestamps, given the observations from the previous L timestamps. A multivariate time series comprises multiple related time series, each representing a different variable. Let $\mathbf{x}_t = [\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \ldots, \mathbf{x}_t^{(M)}] \in \mathbb{R}^{M \times 1}$ be a multivariate signal, where $\mathbf{x}_t^{(i)}$ denotes the i-th variate at time t, for $1 \leq i \leq M$.

In Equation (1), our goal is to develop a model that accurately predicts the values for the next K timestamps based on the recent history spanning L timestamps. Here, L and K are referred to as the look-back window and the prediction horizon, respectively. For a single variable, $\hat{x}_{L+1}^{(i)}, \ldots, \hat{x}_{L+K}^{(i)}$ represents the sequence of future values in the next K time steps for the i th variable, which is the prediction target. In contrast, $x_1^{(i)}, \ldots, x_L^{(i)}$ denote the observed data for the i-th variable over the past L time steps. The function F, parameterized by Φ, leverages these observations to effectively forecast the future sequence values.

$$\hat{x}_{L+1}^{(i)}, \ldots, \hat{x}_{L+K}^{(i)} = F\left(x_1^{(i)}, \ldots, x_L^{(i)}; \Phi\right) \qquad (1)$$

3.2 Model Structure

The architecture of AdaMixT is illustrated in Figure 2. The model leverages GPMs (e.g., GPT2 [Radford *et al.*, 2019] and Llama [Touvron *et al.*, 2023]) and DSMs (e.g., PatchTST [Nie *et al.*, 2022]) as experts to efficiently extract multi-scale features from time series, thereby achieving accurate prediction objectives. The entire framework consists of three core modules: Multi-Scale Feature Extraction, Expert Pool, and Adaptive Weighted Gating Network (AWGN).

As detailed in Algorithm 1, AdaMixT adopts a channel-independent framework to process each variable. The process begins with instance normalization to standardize each variable. Following this, the time series is segmented into patches of varying lengths using a patch-based tokenization, enabling the capture of multi-scale temporal features spanning both short-term and long-term dependencies. These patches are subsequently processed by multiple experts, including GPMs for general feature representation and DSMs tailored for time-series forecasting. Each expert extracts features from input patches and integrates their outputs based on the scoring results of Adaptive Weighted Gating Network. The fused results are then passed through Linear Head to generate the final predictions. Furthermore, while AdaMixT inherently employs a channel-independent design, its architecture is highly flexible and can be seamlessly extended to other patch-based time series models, such as Crossformer [Zhang and Yan, 2023], for modeling inter-variable dependencies. This versatility establishes AdaMixT as an efficient

Algorithm 1 Forcasting Process of AdaMixT

Input: Multivariate time series $\mathbf{X}_{1:L}$ with M variables, Look-back window L, Prediction horizon K

Parameters: Learning rate η, Batch size B, Patch length P, Stride S, Scale factors $\{F_1, F_2, \ldots, F_n\}$

Output: Predictions $\hat{\mathbf{X}}_{L+1:L+K}$

```
for $i \leftarrow 1$ to $M$ do
    $x\_norm_{1:L}^{(i)} \leftarrow$ Instance_Normalization($x_{1:L}^{(i)}$)
end for
for $i \leftarrow 1$ to $M$ do
    for $j \leftarrow 1$ to $n$ do
        $p_j^{(i)} \leftarrow$ MF_Extract($x\_norm_{1:L}^{(i)}, P \cdot F_j, S \cdot F_j$)
    end for
end for
for $i \leftarrow 1$ to $M$ do
    for $j \leftarrow 1$ to $n$ do
        $E_j^{(i)} \leftarrow$ Expert_Model$_j(p_j^{(i)})$
    end for
    $G^{(i)} \leftarrow$ Gating_Network($x\_norm_{1:L}^{(i)}$)
    $fused\_feature^{(i)} \leftarrow \sum_{j=1}^{n} G_j^{(i)} \cdot E_j^{(i)}$
end for
for $i \leftarrow 1$ to $M$ do
    $\hat{x}_{L+1:L+K}^{(i)} \leftarrow$ Linear_Head($fused\_feature^{(i)}$)
end for
return $\hat{\mathbf{X}}_{L+1:L+K}$
```

and generalizable solution for time series forecasting tasks.

Forward Process. We denote the i-th univariate sequence of length L as $\mathbf{x}_{1:L}^{(i)} = (\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_L^{(i)})$, where $i = 1, \ldots, M$. The input sequence $(\mathbf{x}_1, \ldots, \mathbf{x}_L)$ is divided into M univariate sequences $\mathbf{x}^{(i)} \in \mathbb{R}^{1\times L}$, each independently processed by a model consisting of multiple experts, with each sequence using a different patch length. Finally, different weights are assigned to each expert model, and multi-feature fusion is performed. After passing through the final Linear Head, the corresponding prediction results are returned as $\hat{\mathbf{x}}^{(i)} = (\hat{\mathbf{x}}_{L+1}^{(i)}, \ldots, \hat{\mathbf{x}}_{L+K}^{(i)}) \in \mathbb{R}^{1\times K}$.

Multi-scale Feature Extraction. Time segments composed of multiple consecutive timestamps are essential for learning effective predictive representations [Nie *et al.*, 2022]. Building on this idea, we incorporate multi-scale features of time series into the modeling process by dividing each input univariate time series $\mathbf{x}_{1:L}^{(i)}$ into patches of varying lengths, which may be overlapping or non-overlapping. Specifically, we define the base patch length P and stride S, and utilize $Scale\_factors = \{F_1, F_2, \ldots, F_n\}$ to adjust the scale sizes, where n denotes the total number of defined scales. Through this patching process, n patch sequences $\mathbf{x}_{1:L}^{(i)(j)} \in \mathbb{R}^{P_j \times N_j}$ are generated, where $N_j = \left\lfloor \frac{L-P_j}{S_j} \right\rfloor + 2$ represents the number of patches derived from the j-th patching operation. Here, $P_j = P \cdot F_j$ denotes the adjusted patch length, and $S_j = S \cdot F_j$ represents the adjusted stride at the j-th scale. To ensure alignment, the original sequence $\mathbf{x}_{1:L}^{(i)}$ is padded at the end with S_j repeated values before performing the patching process.

By adopting patch lengths of varying scales, smaller P_j values enable the $\mathbf{x}_{1:L}^{(i)(j)}$ branch to specifically focus on short-term temporal features with finer granularity, thereby achieving high-resolution modeling. In contrast, larger P_j values are better suited for effectively capturing long-term seasonal variations and trend characteristics, thus achieving the goal of multi-scale feature extraction.

Expert Pool. We propose the concept of an Expert Pool, designed to integrate the strengths of different models to enhance the performance of multivariate time series forecasting. The Expert Pool consists of two core types of models: General Pre-trained Models and Domain-Specific Models. GPMs (e.g., GPT-2 [Radford *et al.*, 2019] and Llama [Touvron *et al.*, 2023]) demonstrate exceptional performance in understanding complex time series data due to their strong generalizability and rich feature representation learning. In contrast, DSMs (e.g., PatchTST [Nie *et al.*, 2022]) focus on precise pattern extraction in time series data, capturing fine-grained features and effectively compensating for the limitations of GPMs in specific tasks.

Unlike existing methods [Jin *et al.*, 2023; Wang *et al.*, 2023] that typically utilize either GPM or DSM individually, our approach combines these two types of models, significantly improving prediction performance through the synergistic effects. Both types of models are based on the Transformer architecture, whose core mechanism is the Attention Mechanism. Specifically, patches are first projected into the Transformer latent space of dimension D using a trainable linear projection $\mathbf{W}_p \in \mathbb{R}^{D\times P}$, and a learnable positional encoding $\mathbf{W}_{\text{pos}} \in \mathbb{R}^{D\times N}$ is added to preserve temporal order. The transformed patches are then processed by the multi-head attention module, where **Query (Q)**, **Key (K)**, and **Value (V)** matrices are computed as follows: $\mathbf{Q}_h^{(i)} = \left(\mathbf{x}_d^{(i)}\right)^T \mathbf{W}_h^Q$, $\mathbf{K}_h^{(i)} = \left(\mathbf{x}_d^{(i)}\right)^T \mathbf{W}_h^K$, $\mathbf{V}_h^{(i)} = \left(\mathbf{x}_d^{(i)}\right)^T \mathbf{W}_h^V$, where $\mathbf{W}_h^Q, \mathbf{W}_h^K \in \mathbb{R}^{D\times d_k}$ and $\mathbf{W}_h^V \in \mathbb{R}^{D\times D}$. The attention output $\mathbf{O}_h^{(i)} \in \mathbb{R}^{D\times N}$: is then computed using a scaled dot-product: $\left(\mathbf{O}_h^{(i)}\right)^T = \text{Softmax}\left(\frac{\mathbf{Q}_h^{(i)}\mathbf{K}_h^{(i)T}}{\sqrt{d_k}}\right)\mathbf{V}_h^{(i)}$.

Adaptive Weighted Gating Network. In this study, we proposed AWGN, which differs from traditional simple feature fusion methods such as feature addition or concatenation. This innovative approach effectively considers the importance of different features, thereby enhancing the accuracy of time series forecasting. As shown in Figure 2(b), AWGN is a three-layer MLP network that dynamically assigns weights $G(x^{(i)})$ to the expert models based on the characteristics of each input sequence $x_{(i)}$. These weights are subsequently applied to the output features $E(p^{(i)})$ of the expert models, as described in Equation (2). Here, $x^{(i)}$ represents the time series features before patch partitioning, $p^{(i)}$ denotes the sequence after partitioning, and n indicates the total number of experts in the expert model pool.

$$y = \sum_{i=1}^{n} G(x^{(i)}) \cdot E(p^{(i)}) \tag{2}$$

Dataset	# Features	# Timesteps	Frequency	Domain
Weather	21	52696	10 min	Weather
Traffic	862	17544	1 hour	Transportation
Electricity	321	26304	1 hour	Electricity
ILI	7	966	1 week	Illness
ETTh1	7	17420	1 hour	Temperature
ETTh2	7	17420	1 hour	Temperature
ETTm1	7	69680	15 min	Temperature
ETTm2	7	69680	15 min	Temperature

Table 1: Statistics of datasets in various domains.

Finally, the features weighted by AWGN are aggregated and passed through Linear Head to generate the final prediction values. This scoring and weighted aggregation mechanism offers a more efficient approach to integrating multi-feature information for time series forecasting tasks.

Loss Function. We utilize the Mean Squared Error (MSE) loss to quantify the discrepancy between the predictions and the ground truth. The loss in each channel is gathered and averaged over M time series to get the overall objective loss.

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_x \frac{1}{M} \sum_{i=1}^{M} \left\| \hat{x}^{(i)}_{t:t+K-1} - x^{(i)}_{t:t+K-1} \right\|_2^2 \tag{3}$$

Instance Normalization. This technique has been recently introduced to address the distribution shift between training and testing data [Ulyanov *et al.*, 2016]. It works by normalizing each time series instance $x^{(i)}$ to have a zero mean and unit standard deviation. Specifically, each $x^{(i)}$ is normalized before patching, and the mean and standard deviation are added back to the output prediction afterward.

4 Experiments

4.1 Experimental Setup

Dataset. We conduct experiments on eight widely recognized benchmarks, namely Weather, Traffic, Electricity, ILI, and four ETT datasets (ETTh1, ETTh2, ETTm1, and ETTm2), which were originally introduced by [Wu *et al.*, 2021]. The statistics are summarized in Table 1.

Baselines. We evaluate AdaMixT against ten SOTA baselines across four categories: (i) LLM-based models, including TIME-LLM [Jin *et al.*, 2023] and GPT4TS [Zhou *et al.*, 2023]; (ii) Multi-scale time series models, such as TimeMixer [Wang *et al.*, 2024], MICN [Wang *et al.*, 2023], and TimesNet [Wu *et al.*, 2022]; (iii) Transformer-based models, including iTransformer [Liu *et al.*, 2023], PatchTST [Nie *et al.*, 2022], FEDformer [Zhou *et al.*, 2022], and Autoformer [Wu *et al.*, 2021]; and (iv) the MLP-based model, DLinear [Zeng *et al.*, 2023].

Implementation Details. To ensure the fairness of experiments, all baseline models are configured according to the settings in their official open-source repositories and evaluated using a unified evaluation framework, TSLib [1]. All experiments are implemented based on PyTorch [Paszke, 2019] and executed on an NVIDIA A100-80G GPU. The default optimizer for the experiments is Adam [Kingma and Ba, 2014], and each experiment is repeated three times, with the average results reported.

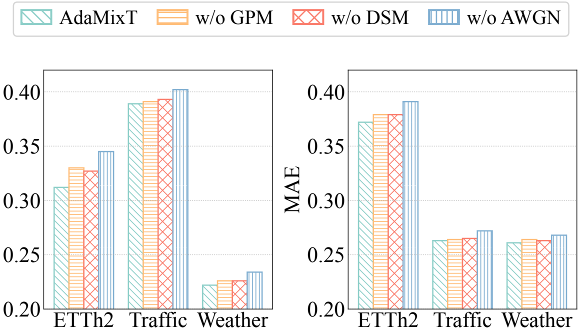


Figure 3: Average MSE and MAE comparison with different model variants on ETTh2, Traffic, and Weather datasets.

4.2 Performance Comparison

Table 2 presents the experimental results for long-term multivariate forecasting, highlighting the substantial advantages of our model. Specifically, compared to LLM-based methods, our model reduces MSE and MAE by 13.19% and 5.35%, respectively. When compared to Multi-scale models, the reductions are 24.99% and 15.14%, respectively. Relative to the DLinear model, MSE and MAE decrease by 22.73% and 13.35%. Furthermore, our model exhibits even more significant improvements over other transformer-based models. Notably, even when compared with SOTA models such as TIME-LLM [Jin *et al.*, 2023] and PatchTST [Nie *et al.*, 2022], our model consistently delivers superior performance. These results strongly demonstrate the robustness and superiority of our approach across various datasets.

4.3 Model Analysis

Ablation Study. To comprehensively evaluate the contributions of the key components—GPM, DSM, and AWGN—to the overall performance of the AdaMixT, we conducted a series of ablation experiments. These experiments systematically removed each component, allowing us to compare the performance of the resulting model variants and quantify the impact of each element.

As shown in Figure 3, the ablation study results demonstrate that the complete AdaMixT model achieves the best performance on the ETTh2, Weather, and Traffic datasets, while removing any single module leads to a decrease in prediction accuracy. For example, on the ETTh2 dataset, removing GPM and DSM results in performance degradation, highlighting the critical role of GPM in extracting general feature representations and the significant value of DSM in modeling temporal features. Notably, the most substantial performance drop is observed when AWGN is removed, with both MSE and MAE significantly higher than other variants. This finding underscores the pivotal role of AWGN in multi-scale feature fusion and its importance in improving prediction accuracy. Similar trends are observed on the Weather and Traffic

[1] https://github.com/thuml/Time-Series-Library

Method		AdaMixT (**Ours**)		TIME-LLM (2023)		GPT4TS (2023)		TimeMixer (2024)		MICN (2023)		TimesNet (2022)		iTransformer (2023)		PatchTST (2022)		FEDformer (2022)		Autoformer (2021)		DLinear (2023)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.145**	**0.196**	0.149	0.200	0.162	0.212	0.161	0.209	0.161	0.229	0.172	0.220	0.175	0.216	0.149	0.198	0.238	0.314	0.249	0.329	0.176	0.238
	192	**0.190**	**0.238**	0.193	0.243	0.204	0.248	0.207	0.250	0.220	0.281	0.219	0.261	0.225	0.258	0.194	0.241	0.275	0.329	0.325	0.370	0.218	0.277
	336	**0.243**	**0.279**	**0.243**	0.284	0.254	0.286	0.264	0.292	0.278	0.331	0.280	0.306	0.280	0.298	0.245	0.282	0.339	0.377	0.351	0.391	0.262	0.313
	720	**0.310**	**0.332**	0.315	0.336	0.326	0.337	0.344	0.343	0.311	0.356	0.365	0.359	0.361	0.351	0.314	0.334	0.389	0.409	0.415	0.426	0.327	0.367
	Avg	**0.222**	**0.261**	0.225	0.266	0.237	0.271	0.244	0.274	0.243	0.299	0.259	0.287	0.260	0.281	0.226	0.264	0.310	0.357	0.335	0.379	0.246	0.299
Traffic	96	**0.358**	**0.248**	0.376	0.280	0.388	0.282	0.466	0.293	0.519	0.309	0.593	0.321	0.394	0.269	0.360	0.249	0.576	0.359	0.597	0.371	0.413	0.288
	192	**0.378**	**0.254**	0.397	0.294	0.407	0.290	0.507	0.301	0.537	0.315	0.617	0.336	0.412	0.277	0.379	0.256	0.610	0.380	0.607	0.382	0.423	0.287
	336	**0.390**	**0.263**	0.420	0.311	0.412	0.294	0.525	0.309	0.534	0.313	0.629	0.336	0.425	0.283	0.392	0.264	0.608	0.375	0.623	0.387	0.438	0.300
	720	**0.428**	0.287	0.448	0.326	0.450	0.312	0.552	0.325	0.577	0.325	0.640	0.350	0.460	0.301	0.432	**0.286**	0.621	0.375	0.639	0.395	0.466	0.315
	Avg	**0.389**	**0.263**	0.410	0.303	0.414	0.295	0.513	0.307	0.542	0.316	0.620	0.336	0.423	0.283	0.391	0.264	0.604	0.372	0.617	0.384	0.435	0.298
Electricity	96	**0.118**	**0.214**	0.137	0.244	0.139	0.238	0.120	0.215	0.164	0.269	0.168	0.272	0.148	0.240	0.129	0.222	0.186	0.302	0.196	0.313	0.141	0.240
	192	**0.146**	**0.237**	0.158	0.266	0.153	0.251	0.170	0.261	0.177	0.285	0.184	0.289	0.164	0.256	0.147	0.240	0.197	0.311	0.211	0.324	0.158	0.260
	336	**0.160**	**0.258**	0.183	0.292	0.169	0.266	0.187	0.278	0.193	0.304	0.198	0.300	0.178	0.271	0.163	0.259	0.213	0.328	0.214	0.327	0.171	0.271
	720	**0.194**	**0.289**	0.247	0.348	0.206	0.297	0.228	0.313	0.212	0.321	0.220	0.320	0.211	0.300	0.197	0.290	0.233	0.344	0.236	0.342	0.206	0.304
	Avg	**0.155**	**0.250**	0.181	0.288	0.167	0.263	0.176	0.267	0.187	0.295	0.193	0.295	0.175	0.267	0.159	0.253	0.207	0.321	0.214	0.327	0.169	0.269
ILI	24	1.384	0.757	1.708	0.765	2.063	0.881	1.358	0.763	2.684	1.112	2.317	0.934	1.638	0.831	**1.319**	**0.754**	2.624	1.095	2.906	1.182	1.964	0.975
	36	**1.300**	**0.755**	1.634	0.781	1.868	0.892	1.432	0.826	2.667	1.068	1.972	0.920	1.742	0.879	1.430	0.834	2.516	1.021	2.585	1.038	2.080	0.998
	48	**1.475**	0.793	1.597	**0.769**	1.790	0.884	1.551	0.814	2.558	1.052	2.238	0.940	1.826	0.932	1.553	0.815	2.505	1.041	3.024	1.145	2.163	1.043
	60	**1.460**	0.821	1.565	**0.754**	1.979	0.957	1.614	1.827	2.747	1.110	2.027	0.928	1.954	0.973	1.470	0.788	2.742	1.122	2.761	1.114	2.396	1.112
	Avg	**1.405**	0.782	1.626	**0.767**	1.925	0.904	1.489	1.058	2.664	1.086	2.139	0.931	1.790	0.904	1.443	0.798	2.597	1.070	2.819	1.120	2.151	1.032
ETTh1	96	**0.360**	**0.393**	0.398	0.414	0.376	0.397	0.381	0.398	0.421	0.431	0.384	0.402	0.386	0.405	0.370	0.399	0.376	0.415	0.435	0.446	0.422	0.448
	192	**0.398**	**0.418**	0.442	0.440	0.416	**0.418**	0.442	0.430	0.474	0.487	0.436	0.429	0.441	0.436	0.413	0.421	0.423	0.446	0.456	0.457	0.419	0.430
	336	**0.398**	**0.427**	0.456	0.450	0.442	0.433	0.501	0.460	0.770	0.672	0.521	0.500	0.491	0.461	0.422	0.436	0.444	0.462	0.486	0.487	0.460	0.462
	720	0.453	0.465	0.602	0.545	0.477	**0.456**	0.544	0.505	0.770	0.672	0.493	0.505	0.509	0.494	**0.447**	0.466	0.469	0.492	0.515	0.517	0.521	0.500
	Avg	**0.402**	**0.426**	0.475	0.462	0.428	**0.426**	0.467	0.448	0.559	0.535	0.458	0.450	0.457	0.449	0.413	0.431	0.428	0.454	0.473	0.477	0.449	0.461
ETTh2	96	**0.260**	**0.328**	0.309	0.362	0.285	0.342	0.288	0.340	0.299	0.364	0.340	0.374	0.300	0.350	0.274	0.336	0.332	0.374	0.332	0.368	0.279	0.344
	192	**0.306**	**0.370**	0.362	0.395	0.354	0.389	0.391	0.403	0.441	0.454	0.402	0.414	0.382	0.400	0.339	0.379	0.407	0.446	0.426	0.434	0.361	0.401
	336	**0.306**	**0.372**	0.376	0.409	0.373	0.407	0.422	0.427	0.654	0.567	0.452	0.452	0.424	0.432	0.329	0.380	0.400	0.447	0.477	0.479	0.466	0.473
	720	**0.376**	**0.418**	0.405	0.436	0.406	0.441	0.442	0.451	0.956	0.716	0.462	0.468	0.426	0.445	0.379	0.422	0.412	0.469	0.453	0.490	0.398	0.417
	Avg	**0.312**	**0.372**	0.363	0.401	0.355	0.395	0.386	0.405	0.588	0.525	0.414	0.427	0.383	0.407	0.330	0.379	0.388	0.434	0.422	0.443	0.351	0.409
ETTm1	96	**0.288**	**0.341**	**0.288**	0.343	0.292	0.346	0.317	0.358	0.316	0.362	0.338	0.375	0.341	0.376	0.290	0.342	0.326	0.390	0.510	0.492	0.303	0.346
	192	**0.328**	0.369	0.347	0.378	0.332	0.372	0.367	0.387	0.363	0.390	0.374	0.387	0.381	0.395	0.332	0.369	0.365	0.415	0.514	0.495	0.338	**0.367**
	336	**0.359**	**0.388**	0.368	0.394	0.366	0.394	0.388	0.402	0.408	0.426	0.410	0.411	0.419	0.419	0.366	0.392	0.392	0.425	0.510	0.492	0.375	0.393
	720	**0.415**	**0.418**	0.421	0.423	0.417	0.421	0.454	0.443	0.481	0.476	0.478	0.450	0.486	0.456	0.416	0.420	0.446	0.458	0.527	0.493	0.427	0.422
	Avg	**0.348**	**0.379**	0.356	0.385	0.352	0.383	0.382	0.398	0.392	0.414	0.400	0.406	0.407	0.412	0.351	0.381	0.382	0.422	0.515	0.493	0.361	0.382
ETTm2	96	**0.163**	**0.252**	0.168	0.257	0.173	0.262	0.175	0.259	0.179	0.275	0.187	0.267	0.184	0.267	0.165	0.255	0.180	0.271	0.205	0.293	0.165	0.257
	192	**0.217**	**0.290**	0.219	0.293	0.229	0.301	0.237	0.299	0.307	0.376	0.249	0.309	0.253	0.312	0.220	0.292	0.252	0.318	0.278	0.336	0.227	0.307
	336	**0.272**	0.330	0.275	0.332	0.286	0.341	0.296	0.338	0.325	0.388	0.321	0.351	0.315	0.352	0.274	**0.329**	0.324	0.364	0.343	0.379	0.285	0.342
	720	**0.361**	0.383	0.367	**0.335**	0.378	0.401	0.393	0.395	0.502	0.490	0.408	0.403	0.412	0.406	0.362	0.385	0.410	0.420	0.414	0.419	0.398	0.417
	Avg	**0.253**	0.314	0.257	**0.304**	0.267	0.326	0.275	0.323	0.328	0.382	0.291	0.333	0.291	0.334	0.255	0.315	0.292	0.343	0.310	0.357	0.269	0.331
1^{st} Count		**38**	**30**	2	5	0	3	0	0	0	0	0	0	0	0	2	3	0	0	0	0	0	1

Table 2: Long-term forecasting results on different datasets. "Avg" is the average of all considered prediction lengths. Lower MSE/MAE indicates better performance. We use prediction lengths $K \in \{24, 36, 48, 60\}$ for ILI and $K \in \{96, 192, 336, 720\}$ for the others. The best and second-best results are marked in **bold** and underlined, respectively.

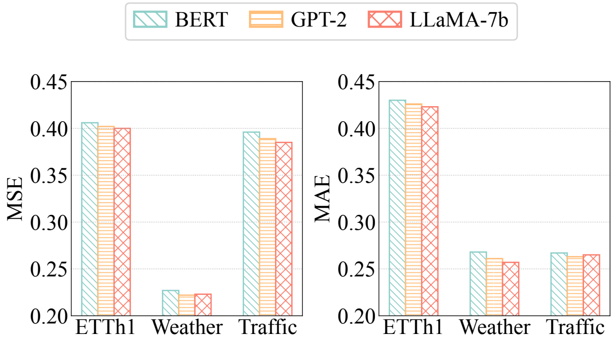


Figure 4: Multivariate long-term forecasting results with different pretrained models in AdaMixT.

datasets, further confirming the importance of each module in enhancing model performance.

Results with Different Models. In the main experiments, AdaMixT utilizes GPT-2 [Radford *et al.*, 2019] as GPM. To further validate the effectiveness of the model, we also perform comparative experiments using BERT and LLaMA-7B. The experimental results in Figure 4 indicate that the performance differences among BERT, GPT-2, and LLaMA-7B are minimal on the ETTh1, Weather, and Traffic datasets. This finding demonstrates the high robustness of AdaMixT in selecting GPMs, as it consistently delivers stable performance across various pretrained models.

Hyperparameter Sensitivity. We conducte a sensitivity analysis on 3 key hyperparameters, including the number of layers in the backbone model, the look-back window, and the number of experts. Results are shown in Figure 5. Based on the results, we summarize the following conclusions:

- **Backbone Layers**: The number of layers in the backbone is positively correlated with the predictive performance. This indicates that, even after fusion, GPM retain favorable scaling laws with respect to layer depth, which positively influences model performance.
- **Look-back Window**: The Look-back window directly affects prediction accuracy, especially at extended forecasting horizons. This observation aligns with the patterns

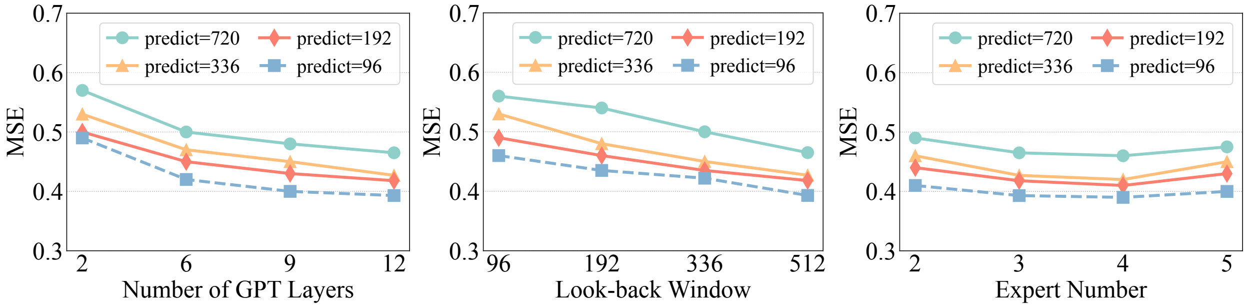


Figure 5: Sensitivity analysis of three key hyperparameters on the ETTh1 dataset.

Method		AdaMixT(1, 1/2)		AdaMixT(1, 1)		AdaMixT(1, 2)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ILI	24	1.702	0.854	1.630	0.824	**1.384**	**0.757**
	36	1.753	0.862	1.509	0.816	**1.300**	**0.755**
	48	1.737	0.860	2.023	0.941	**1.475**	**0.793**
	60	1.560	0.839	1.630	0.891	**1.460**	**0.821**
ETTh1	96	**0.360**	**0.393**	0.366	0.399	0.381	0.408
	192	**0.398**	**0.418**	0.400	0.415	0.412	0.423
	336	**0.398**	**0.427**	0.411	0.431	0.407	0.428
	720	**0.453**	**0.465**	0.459	0.470	0.571	0.547

Table 3: Impact of different feature scales on prediction accuracy for ILI and ETTh1 datasets. The best results are marked in **bold**.

found in traditional models, demonstrating that longer historical inputs can effectively enhance performance.

- **Number of Experts**: The experimental results indicate that increasing the number of expert models can effectively capture features at different scales, thereby significantly improving the accuracy of time series forecasting. This result validates the effectiveness of the feature fusion mechanism in the AdaMixT. However, caution should be taken as the model may be prone to overfitting.

Scale Factors Study. As shown in Algorithm 1, the setting of scale factors determines the granularity of feature extraction. The selection of the appropriate parameters is critical to the accuracy of the prediction. To further investigate this, we conduct an analysis of the impact of different feature scale settings on prediction performance, using the ILI and ETTh1 datasets as examples.

Through a cyclic study of these two datasets, we find that the cyclicity of ILI is significantly longer than that of ETTh1. Based on this observation, we employ two expert models in the experiment, with the scale factor for GPM set to 1, and the scale factor range for the DSM set to $\{1/2, 1, 2\}$. The results, as shown in Table 3, indicate that for datasets with longer periodicities, using larger scale factors improves prediction performance (and vice versa). This finding suggests that when selecting scale factors, the intrinsic characteristics of the time series should be considered to achieve optimal prediction results.

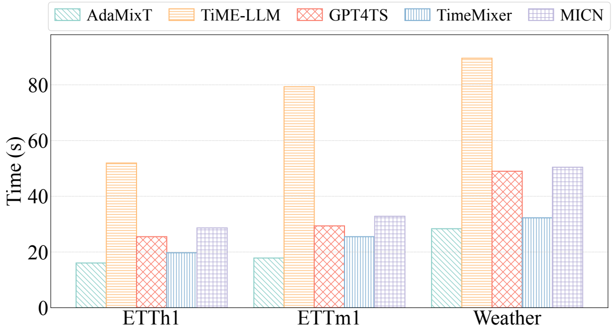


Figure 6: Comparison of inference time for different models across ETTh1, ETTm1, and Weather datasets.

Inference Time Study. In order to evaluate the practical applicability, we compare the inference time of AdaMixT with current similar methods. As shown in Figure 6, AdaMixT demonstrates superior inference time compared with other complex multi-scale feature fusion models and LLM-based models. This advantage is primarily attributed to the fact that AdaMixT does not require complex prompt generation, post-processing, or frequency domain transformations.

5 Conclusion and Future Works

In this paper, we present AdaMixT, which is designed to address the limitations of existing methods in terms of generalizability and multi-scale feature fusion. AdaMixT incorporates three key innovations: Multi-scale Feature Extraction, Expert Pool, and Adaptive Weighted Gating Network, significantly enhancing the model's performance. To the best of our knowledge, this work is the first to combine the general feature representation capability of GPM with the fine-grained modeling ability of DSM to construct the expert pool. Additionally, we design AWGN that evaluates the weights of different models, enabling effective fusion of multi-scale features. Extensive experimental results on multiple benchmark datasets demonstrate that our approach outperforms existing forecasting approaches.

Our model offers new insights into multi-scale feature fusion in time series analysis. In the future, we aim to further explore multi-scale fusion methods and strengthen the complementary strengths of GPM and DSM, which will be a crucial research direction moving forward.

Acknowledgments

This work is supported by the National Key Laboratory under grant 241-HF-Z01-01. Guoliang Li is supported by the National Key R&D Program of China (2023YFB4503600), NSF of China (62232009), Zhongguancun Lab, Huawei, and Beijing National Research Center for Information Science and Technology (BNRist).

References

[Aljundi *et al.*, 2017] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pages 3366–3375, 2017.

[Chen *et al.*, 2024] Minxiao Chen, Haitao Yuan, Nan Jiang, Zhifeng Bao, and Shangguang Wang. Urban traffic accident risk prediction revisited: Regionality, proximity, similarity and sparsity. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 281–290, 2024.

[Chen *et al.*, 2025] Minxiao Chen, Haitao Yuan, Nan Jiang, Zhihan Zheng, Sai Wu, Ao Zhou, and Shangguang Wang. RLOMM: an efficient and robust online map matching framework with reinforcement learning. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 209:1–209:26, 2025.

[Das and Dutta, 2020] Sourya Dipta Das and Saikat Dutta. Fast deep multi-patch hierarchical network for nonhomogeneous image dehazing. In *IEEE/CVF Computer Vision and Pattern Recognition Conference Workshops (CVPRW)*, pages 482–483, 2020.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[He *et al.*, 2024a] Yuanpeng He, Yali Bi, Lijian Li, Chi-Man Pun, Wenpin Jiao, and Zhi Jin. Mutual evidential deep learning for semi-supervised medical image segmentation. In *IEEE InternationalConference onBioinformatics and Biomedicine (BIBM)*, pages 2010–2017, 2024.

[He *et al.*, 2024b] Yuanpeng He, Lijian Li, Tianxiang Zhan, Wenpin Jiao, and Chi-Man Pun. Generalized uncertainty-based evidential fusion with hybrid multi-head attention for weak-supervised temporal action localization. In *IEEE InternationalConference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3855–3859, 2024.

[Jacobs *et al.*, 1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural Computation*, pages 79–87, 1991.

[Jiang *et al.*, 2024] Nan Jiang, Haitao Yuan, Jianing Si, Minxiao Chen, and Shangguang Wang. Towards effective next POI prediction: Spatial and semantic augmentation with remote sensing data. In *IEEE International Conference on Data Engineering (ICDE)*, pages 5061–5074, 2024.

[Jin *et al.*, 2023] 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-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lepikhin *et al.*, 2020] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[Li *et al.*, 2022] LinYu Li, Xuan Zhang, YuBin Ma, Chen Gao, Jishu Wang, Yong Yu, Zihao Yuan, and Qiuying Ma. A knowledge graph completion model based on contrastive learning and relation enhancement method. *Knowledge-Based Systems (KBS)*, pages 1–15, 2022.

[Li *et al.*, 2025] Linyu Li, Zhi Jin, Xuan Zhang, Haoran Duan, Jishu Wang, Zhengwei Tao, Haiyan Zhao, and Xiaofeng Zhu. Multi-view riemannian manifolds fusion enhancement for knowledge graph completion. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, pages 2756–2770, 2025.

[Lin *et al.*, 2024a] Jiaye Lin, Qing Li, Guorui Xie, Zhongxu Guan, Yong Jiang, Ting Xu, Zhong Zhang, and Peilin Zhao. Mitigating sample selection bias with robust domain adaption in multimedia recommendation. In *ACM International Conference on Multimedia (MM)*, pages 7581–7590, 2024.

[Lin *et al.*, 2024b] Jiaye Lin, Shuang Peng, Zhong Zhang, and Peilin Zhao. Tlrec: A transfer learning framework to enhance large language models for sequential recommendation tasks. In *ACM Conference on Recommender Systems (RecSys)*, pages 1119–1124, 2024.

[Liu *et al.*, 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. *arXiv preprint arXiv:2310.06625*, 2023.

[Liu *et al.*, 2025] Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 18780–18788, 2025.

[Miao *et al.*, 2024a] Hao Miao, Ziqiao Liu, Yan Zhao, Chenjuan Guo, Bin Yang, Kai Zheng, and Christian S Jensen. Less is more: Efficient time series dataset condensation via two-fold modal matching. *Proceedings of the VLDB Endowment (PVLDB)*, pages 226–238, 2024.

[Miao *et al.*, 2024b] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Kai Zheng, Feiteng Huang, Jiandong Xie, and Christian S Jensen. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *IEEE International Conference on Data Engineering (ICDE)*, pages 1050–1062, 2024.

[Miao *et al.*, 2025] Hao Miao, Ronghui Xu, Yan Zhao, Senzhang Wang, Jianxin Wang, Philip S Yu, and Christian S Jensen. A parameter-efficient federated framework for streaming time series anomaly detection via lightweight adaptation. *IEEE Transactions on Mobile Computing (TMC)*, pages 1–14, 2025.

[Nie *et al.*, 2022] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

[Paszke, 2019] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, pages 1–24, 2019.

[Shazeer *et al.*, 2017] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[Ulyanov *et al.*, 2016] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[Wang *et al.*, 2023] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *International Conference on Learning Representations (ICLR)*, pages 1–22, 2023.

[Wang *et al.*, 2024] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Conference on Neural Information Processing Systems (NeurIPS)*, 34:22419–22430, 2021.

[Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

[Yuan and Li, 2021] Haitao Yuan and Guoliang Li. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. In *Data Science and Engineering (DSE)*, pages 63–85, 2021.

[Yuan *et al.*, 2020] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. Effective travel time estimation: When historical trajectories over road networks matter. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 2135–2149, 2020.

[Yuan *et al.*, 2021] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. An effective joint prediction model for travel demands and traffic flows. In *IEEE International Conference on Data Engineering (ICDE)*, pages 348–359, 2021.

[Yuan *et al.*, 2023] Haitao Yuan, Sai Wang, Zhifeng Bao, and Shangguang Wang. Automatic road extraction with multi-source data revisited: completeness, smoothness and discrimination. *Proceedings of the VLDB Endowment (PVLDB)*, pages 3004–3017, 2023.

[Yuan *et al.*, 2024] Haitao Yuan, Gao Cong, and Guoliang Li. Nuhuo: An effective estimation model for traffic speed histogram imputation on A road network. *Proceedings of the VLDB Endowment (PVLDB)*, pages 1605–1617, 2024.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 11121–11128, 2023.

[Zhang and Yan, 2023] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations (ICLR)*, pages 1–21, 2023.

[Zhao *et al.*, 2024] Zhengyang Zhao, Haitao Yuan, Nan Jiang, Minxiao Chen, Ning Liu, and Zengxiang Li. STMGF: an effective spatial-temporal multi-granularity framework for traffic forecasting. In *International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 235–245, 2024.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 11106–11115, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning (ICML)*, pages 27268–27286, 2022.

[Zhou *et al.*, 2023] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Conference on Neural Information Processing Systems (NeurIPS)*, pages 43322–43355, 2023.