Many Minds, One Goal: Time Series Forecasting via Sub-task Specialization and Inter-agent Cooperation

Qihe Huang[†], Zhengyang Zhou[†,§,✉], Yangze Li[†], Kuo Yang[†], Binwu Wang[†,§],
Yang Wang[†,§,✉]

[†] University of Science and Technology of China (USTC), Hefei, China
[§] Suzhou Institute for Advanced Research, USTC, Suzhou, China
Email: {hqh,liyangze,yangkuo}@mail.ustc.edu.cn, {wbw2024, zzy0929, angyan}@ustc.edu.cn

Abstract

Time series forecasting is a critical and complex task, characterized by diverse temporal patterns, varying statistical properties, and different prediction horizons across datasets and domains. Conventional approaches typically rely on a single, unified model architecture to handle all forecasting scenarios. However, such monolithic models struggle to generalize across dynamically evolving time series with shifting patterns. In reality, different types of time series may require distinct modeling strategies. Some benefit from homogeneous multi-scale forecasting awareness, while others rely on more complex and heterogeneous signal perception. Relying on a single model to capture all temporal diversity and structural variations leads to limited performance and poor interpretability. To address this challenge, we propose a Multi-Agent Forecasting System (MAFS) that abandons the one-size-fits-all paradigm. MAFS decomposes the forecasting task into multiple sub-tasks, each handled by a dedicated agent trained on specific temporal perspectives (e.g., different forecasting resolutions or signal characteristics). Furthermore, to achieve holistic forecasting, agents share and refine information through different communication topology, enabling cooperative reasoning across different temporal views. A lightweight voting aggregator then integrates their outputs into consistent final predictions. Extensive experiments across 11 benchmarks demonstrate that MAFS significantly outperforms traditional single-model approaches, yielding more robust and adaptable forecasts. **Code:** `https://github.com/h505023992/MAFS`

1 Introduction

Time series forecasting [79, 78, 49, 7, 2, 3, 52, 11, 13, 12, 53, 64] plays a vital role in a wide range of real-world applications, including finance, energy, healthcare, and intelligent transportation. Despite remarkable progress achieved by deep learning models such as RNNs [9, 24], CNNs [55, 22], and Transformers [60, 28, 23, 76, 25, 76], many existing approaches rely on monolithic architectures that often struggle to generalize across ever-changing temporal patterns and continuously-evolving signal characteristics inherent in time series [27, 16, 29, 51].

Meanwhile, Multi-Agent Systems (MAS) [18, 48] have emerged as a powerful paradigm for addressing complex problems through collaboration among specialized agents [75]. This framework has achieved remarkable success in a wide range of domains, including robotics [66], question answering [65], and sequential decision-making [67]. Crucially, Multi-Agent Systems allow individual agents to process information from distinct perspectives and coordinate their outputs to

✉ Yang Wang and Zhengyang Zhou are corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

achieve a common objective [74, 57]. This design promotes modularity, scalability, and robustness in solving high-dimensional and complex tasks [40, 5]. Inspired by these successes, a natural question arises: *Can Multi-Agent Systems also benefit time series forecasting, particularly in scenarios with heterogeneous temporal patterns or task decomposition requirements?*

However, directly applying multi-agent systems to time series forecasting remains non-trivial and faces three key challenges: ***(i) Task Decomposition for Specialized Modeling.*** Time series forecasting tasks are inherently unified and not easily divisible [8]. *A critical question arises: how can we decompose a global forecasting task into meaningful sub-tasks that enable each agent to develop domain-specific expertise?* Without proper decomposition, agents risk learning overlapping or redundant representations, which limits the benefits of specialization [68]. ***(ii) Limited Agent Perception.*** If an agent consistently operates within a narrow temporal scope, such as focusing only on local trends or periodic components, it may fail to capture important contextual signals needed for precise processing [74, 57]. A lack of holistic understanding can hinder the agent's ability to generalize across variable temporal conditions. ***(iii) Forecasting Collaboration Bottlenecks.*** Even when agents are well-specialized, enabling effective collaboration among them remains a substantial challenge [30, 4]. Agents must not only share information efficiently but also resolve potential conflicts in their predictions. Poor coordination can lead to inconsistent or contradictory outputs, undermining the overall forecasting accuracy [47].

To bridge the gap between monolithic forecasting models and the need for adaptive, collaborative intelligence, we propose a novel **Multi-Agent Forecasting System (MAFS)** that introduces principled mechanisms for modular time series forecasting. Fundamentally, to enable each agent to specialize in modeling a distinct temporal attribute, MAFS explores two types of subtask decomposition: (i) forecasting across multiple *homogeneous future horizons*, and (ii) predicting *heterogeneous signal features* such as frequency-domain energy, statistical moments, periodicity, and trend. This design allows each agent to learn a well-defined aspect of the temporal structure. Furthermore, to expand each agent's receptive field beyond local input views, we introduce **inter-agent communication** that allows information exchange at the representation level. This is implemented via structured message passing over predefined topologies, including *Ring*, *Star*, *Chain*, and *Fully-Connected Graphs*, enabling flexible and scalable coordination across agents. Third, to realize effective decision fusion across agents, MAFS incorporates a **two-stage voting aggregator** composed of: (i) an *Agent Confidence Estimator* that evaluates the confidence or relevance of its own prediction; and (ii) a *Global Voter* that aggregates forecasts across agents based on both their internal ratings and mutual assessments. This mechanism enables robust collaboration by assigning adaptive weights to different agents during inference. Overall, MAFS is designed to fully harness the strengths of collective intelligence while maintaining scalability and generalization across a variety of forecasting scenarios. Our contributions are summarized as follows:

- We propose **Multi-Agent Forecasting System (MAFS)**, the first general-purpose time series forecasting framework based on Multi-Agent Systems, which leverages collective intelligence to tackle complex, evolving, and heterogeneous temporal patterns.
- Within **MAFS**, we introduce two principled task decomposition strategies to enable each agent to specialize in distinct forecasting sub-tasks. Furthermore, we design an inter-agent communication module to enhance the generalization capacity of each agent through structured message passing. Finally, a two-stage voting aggregator combines self-assessments and global voting to facilitate robust and coordinated multi-agent forecasting.
- Through agent specialization and structured collaboration, MAFS demonstrates superior forecasting performance. Extensive experiments on 11 real-world datasets demonstrate that MAFS consistently outperforms competitive baselines, achieving on average a **6.35% reduction in MSE** and a **4.03% reduction in MAE** compared to its single-agent counterpart. MAFS also ranks first on **16 out of 22 metrics**, and secures a top-2 position on **20 out of 22 metrics**, spanning both MSE and MAE evaluations.

2 Related Work

2.1 Time Series Forecasting

Time series forecasting is a foundational task in machine learning that aims to predict future values from past observations [63, 36, 50, 62, 21, 70, 35, 54, 56, 43, 32, 42, 10]. Traditional statistical

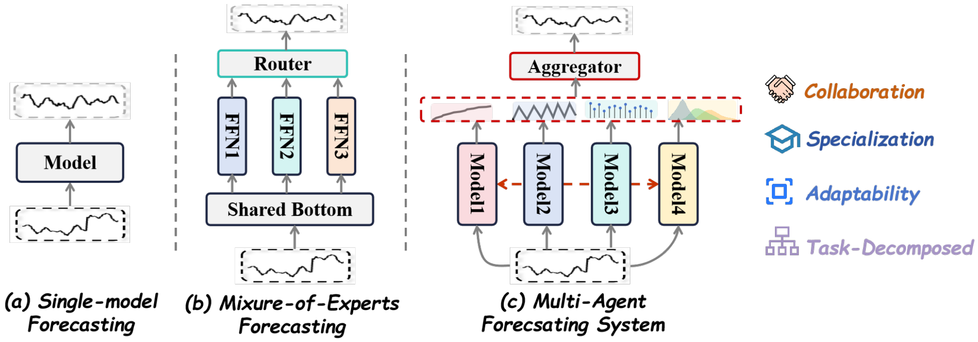

Figure 1: Illustration of different forecasting paradigms: (a) Single-model Forecasting, (b) Mixture-of-Experts Forecasting, and (c) Multi-Agent Forecasting System (MAFS). Compared to traditional paradigms, MAFS achieves better **specialization**, **collaboration**, **adaptability**, and **interpretability**.

models such as ARIMA [1] and VAR [31] are widely used for their simplicity, yet often struggle with capturing nonlinear and long-term dependencies. With the advent of deep learning, models such as RNNs [9, 24], MLPs [58, 37, 29, 34, 14], and Transformers [25, 44, 69, 77, 71, 76, 38, 33] have demonstrated remarkable improvements in modeling complex temporal dynamics. As shown in Figure 1(a), these models typically adopt monolithic architectures, where a single model is trained end-to-end to learn all patterns across the entire time series. However, such one-size-fits-all approaches often underperform in heterogeneous or nonstationary environments, where different segments of the data may exhibit distinct behaviors or require different inductive biases [16].

2.2 Mixture-of-Experts Forecasting

To improve modeling capacity and flexibility, Mixture-of-Experts (MoE) [45] has been applied to time series forecasting, where multiple expert models are trained to extract diverse high-level features from the input sequences. As shown in Figure 1(b), a Router mechanism is typically employed to dynamically select or weight experts for adaptive specialization across different temporal patterns. Time-MoE [46] introduces a scalable autoregressive Transformer equipped with sparse mixture-of-experts, enabling billion-scale time series pretraining with reduced inference cost and flexible forecasting horizons. From frequency aspect, MOIRAI-MoE [26] eliminates the need for human-defined frequency specialization by learning token-level expert routing within a sparse MoE framework. Despite their strengths, MoE-based forecasting models typically operate within a monolithic architecture and lack modular agent-level interpretability [20, 72].

2.3 Multi-Agent System

Multi-agent Systems (MAS) has emerged as a powerful paradigm for solving complex problems through the cooperation of multiple specialized agents [4]. In the context of machine learning, MAS enables agents to learn distinct competencies, share information, and coordinate actions to collectively solve tasks that are difficult for a single model [30]. Recently, there has been growing interest in leveraging large language model agents for multi-agent collaboration in open-domain reasoning [75, 65], but its application to time series forecasting remains largely unexplored. As shown in in Figure 1(c), in contrast to Mixture-of-Experts, which implicitly routes inputs through fixed expert networks without agent awareness or coordination, multi-agent systems offer explicit control, communication, and division of labor. Thus, designing an MAS forecasting system offers a promising pathway to harness specialization and collaboration for forecasting complex time series.

3 Problem Formulation

Time Series Forecasting We consider a standard time series forecasting problem. Given a historical input sequence $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_T] \in \mathbb{R}^{T \times M}$ with T time steps and M variables, the goal is to predict the future series $\mathbf{Y} = [\mathbf{x}_{T+1}, \mathbf{x}_{T+2}, .., \mathbf{x}_{T+L}] \in \mathbb{R}^{L \times M}$, where L is future horizon.

Multi-agent System for Forecasting To improve forecasting performance and interpretability, we introduce a multi-agent forecasting system $\mathbf{S} = \{\mathbf{Agent}_1(\cdot), \mathbf{Agent}_2(\cdot), \ldots, \mathbf{Agent}_N(\cdot)\}$

consisting of N specialized forecasting agents. Each $\mathbf{Agent}_i(\cdot)$ is responsible for solving a distinct sub-task of the forecasting problem. The agents interact and exchange information via a predefined communication topology $\mathbf{G} = \{\mathbf{S}, \mathbf{A}, \mathbf{E}\}$, where $\mathbf{A} \in \{0,1\}^{N \times N}$ is a binary adjacency matrix, and $\mathbf{E} \in \mathbb{R}^{N \times N}$ denotes the edge weights. The agent outputs are aggregated by an Agent-rated Voting Aggregator $\mathrm{AVA}(\cdot)$ to produce the final forecasting result. The overall process is formulated as:

$$\hat{\mathbf{Y}} = \mathrm{AVA}(\mathrm{Comm}(\{\mathbf{Agent}_1(\mathbf{X}), \mathbf{Agent}_2(\mathbf{X}), \ldots, \mathbf{Agent}_N(\mathbf{X}\}; \mathbf{G})) \tag{1}$$

where $\mathrm{Comm}(\cdot; \mathbf{G})$ represents the communication mechanism. Let $\Theta_A = \{\theta_1, \ldots, \theta_N\}$ denote the parameters of all agents, Θ_T the parameters of communication weight (i.e., $\mathbf{E}$), and Θ_P the parameters of the Agent-rated Voting Aggregator. The joint training objective is to minimize the expected mean squared error (MSE) between the predicted and the truth:

$$\min_{\Theta_A, \Theta_T, \Theta_P} \mathbb{E}_{(\mathbf{X},\mathbf{Y}) \sim \mathcal{D}} \left[\|\mathbf{Y} - \hat{\mathbf{Y}}\|_2^2 \right], \tag{2}$$

where $\mathcal{D}$ denotes the empirical data distribution.

4 Methodology

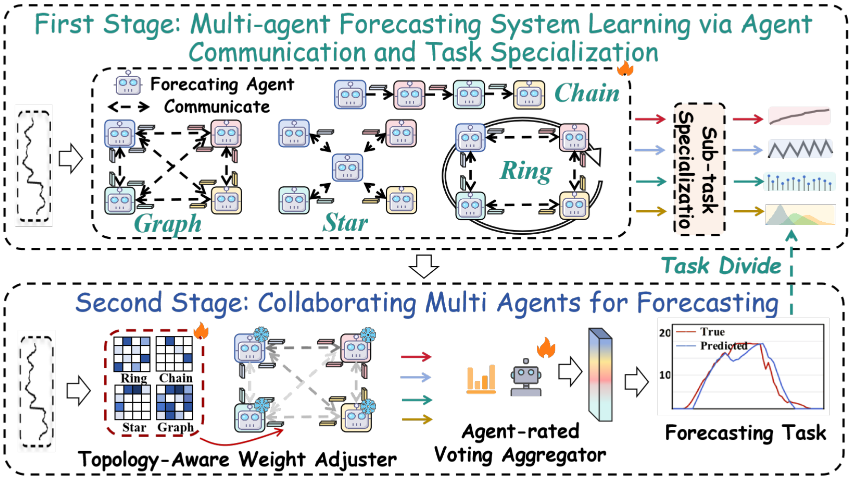
First Stage: Multi-agent Forecasting System Learning via Agent Communication and Task Specialization
Forecating Agent
Communicate
Chain
Graph
Star
Ring
Sub-task Specializatio
Task Divide
Second Stage: Collaborating Multi Agents for Forecasting
Ring
Chain
Star
Graph
Topology-Aware Weight Adjuster
Agent-rated Voting Aggregator
20
10
True
Predicted
Forecasting Task

Figure 2: Overview of the two-stage learning in Multi-Agent Forecasting System.

As illustrated in Figure 2, the Multi-Agent Forecasting System (MAFS) is structured around a two-stage learning paradigm. **Stage 1: Specialization Pretraining.** Each agent is assigned a distinct sub-task (e.g., different signal characteristics) and exchanges hidden states at every layer using a fixed communication graph with uniform weights. Only agent-specific parameters Θ_A are optimized in this stage, enabling each agent to specialize independently. **Stage 2: Collaborative Forecasting.** Agent parameters are frozen. Edge weights in communication is now learnable by Topology-aware Weight Adjuster with Θ_T. An Agent-rated Voting Aggregator (AVA) with parameters Θ_P aggregates the communicated features to produce the final prediction.

4.1 Forecasting Agent Architecture

In the Multi-Agent Forecasting System (MAFS), as shown in Figure 3, each forecasting agent adopts a pluggable encoder-based time series model as its backbone, enabling flexible integration with various architectures. Given an input sequence $\mathbf{X} \in \mathbb{R}^{T \times M}$, the encoding process is formulated as:

$$\mathbf{H}^{(0)} = \mathrm{Embed}(\mathbf{X}^\top), \quad \mathbf{H}^{(l+1)} = \mathrm{EncoderLayer}(\mathbf{H}^{(l)}), \quad \hat{\mathbf{O}} = \mathrm{Head}(\mathbf{H}^{(L)}) \tag{3}$$

Here, $\mathbf{H}^{(l)}$ represents the hidden state at the l-th encoder layer. The $\mathrm{EncoderLayer}(\cdot)$ can follow any standard design, making MAFS extensible to a wide range of forecasting backbones. $\hat{\mathbf{O}}$ is the agent output of specialized forecasting sub-task.

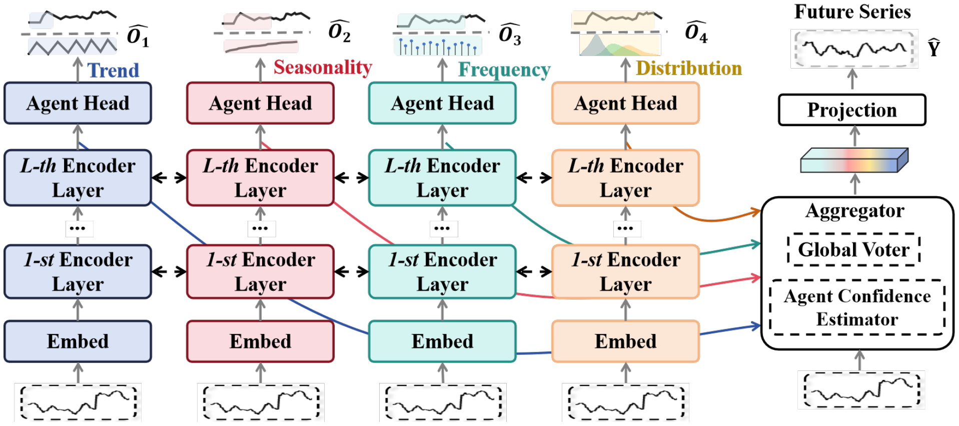

Figure 3: The overall architecture of MAFS. Each forecasting agent employs a pluggable encoder-based time series model, and their representation are aggregated to produce the final forecast.

4.2 Agent Communication

Each agent is assigned with a dedicated forecasting sub-task. While such specialization facilitates inductive bias and modeling diversity, it inevitably leads to data bias. To mitigate this limitation and enable agents to form coherent and globally consistent predictions, we introduce an explicit communication mechanism. This communication can be formally expressed as, $\{\mathbf{HC}_1^l, \mathbf{HC}_2^l, ..., \mathbf{HC}_N^l = \text{Comm}(\mathbf{H}_1^l, \mathbf{H}_2^l, ..., \mathbf{H}_N^l; \mathbf{G}); \mathbf{H}_i^{l+1} = \text{EncoderLayer}_i(\mathbf{HC}_i^l)\}$. Here, $\mathbf{H}_i^l$ denotes the output of the l-th encoder layer of agent i, and $\mathbf{HC}_i^l$ represents the updated representation after communication. The communication function $\text{Comm}(\cdot)$ is instantiated as a graph convolution operation, enabling structured message passing across agents connected via $\mathbf{G}$. Specifically, we implement the communication via a Graph Convolutional Network [19],

$$\mathbf{HC}_i^l = \sigma\left(\sum_{j\in\mathcal{N}(i)} \mathbf{A}_{ij}\cdot\mathbf{W}\cdot\mathbf{H}_j^l\right) \tag{4}$$

where $\mathcal{N}(i)$ is the set of connected agents of $\mathbf{Agent}_i$, $\mathbf{A}_{ij}$ is the normalized adjacency matrix indicating communication strength between agents i and j, $\mathbf{W}\in\mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$ is a learnable projection matrix, and $\sigma(\cdot)$ is a non-linear activation function. This formulation enables each agent to refine its hidden representation by aggregating contextual information from others, thereby overcoming the limitations of isolated modeling and improving coordination among agents.

4.3 Agent Specialization via Sub-task Decomposition

To promote diversity and specialization among agents, we design two sub-task decomposition strategies: **(1) Multi-scale temporal forecasting (homogeneous setting):** Each agent focuses on predicting a different portion of the future horizon at increasing lengths, e.g., the $\mathbf{Agent}_i$ predicts the first $\frac{i}{N}\times L$ steps. This encourages specialization across temporal scales, from short-term to long-term forecasting. **(2) Multi-aspect signal forecasting (heterogeneous setting):** Agents are assigned complementary signal analysis tasks, including trend extraction, seasonality modeling, spectral energy estimation, and statistical summary prediction. Each sub-task targets a distinct property of the future signal, fostering diverse and orthogonal representations. These decompositions enable the agent ensemble to capture rich temporal dynamics from multiple perspectives, improving overall forecasting accuracy. More details are availavle at Appendix A.

4.4 Topology-aware Weight Adjuster

In the second stage of training, we freeze the parameters of each forecasting agent and make the inter-agent communication weights learnable. Specifically, the fixed edge weight matrix $\mathbf{E}$ is replaced by a parameterized matrix $\mathbf{E}_{\Theta_D}\in\mathbb{R}^{N\times N}$, where each entry denotes the learnable communication strength between a pair of agents.

To retain the prior topology, we keep a fixed binary adjacency mask $\mathbf{A} \in 0, 1^{N \times N}$, which encodes the initial traffic-aware structure. The learnable edge weights are modulated by this mask to obtain the soft communication graph, $\mathbf{A}' = \sigma(\mathbf{E}_{\Theta_D}) \odot \mathbf{A}$, where $\sigma(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise multiplication. To ensure symmetric communication [19], we construct the normalized adjacency matrix as,

$$\mathbf{A}_{\text{norm}} = \mathbf{D}^{-1/2}(\hat{\mathbf{A}})\mathbf{D}^{-1/2}, \quad \text{where} \quad \hat{\mathbf{A}} = \frac{1}{2}(\mathbf{A}' + \mathbf{A}'^{\top}) + \mathbf{I}, \quad \mathbf{D} = \text{diag}\left(\sum_j \hat{A}_{ij}\right). \tag{5}$$

This topology-aware normalization allows information exchange to be dynamically adjusted while preserving structural priors. Crucially, $\mathbf{E}_{\Theta_D}$ is optimized jointly with the forecasting objective via backpropagation, enabling the system to learn an adaptive, task-specific communication topology that enhances coordination among agents and improves overall prediction performance.

4.5 Agent-rated Voting Aggregator

During forecasting phase, we aggregate the final embedding from all agents to complete forecasting. To dynamically make decision-making process adaptive to each agent, we design a two-stage voting aggregator (AVA), which include an **Agent Confidence Estimator** and a **Global Voter**.

Agent Confidence Estimator Let $\{\mathbf{H}_1^L, \mathbf{H}_2^L, \dots, \mathbf{H}_N^L\}$ denote the final output representations of all forecasting agents. The Agent Confidence Estimator evaluates the confidence of each agent through a self-assessment mechanism. Each $\mathbf{H}_i^L$ is paired with a shared contextual embedding $\mathbf{C}_i$, and passed through a learnable gating network to produce an element-wise gate coefficient:

$$\alpha_i = \sigma(\text{Gate}([\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_N])) \tag{6}$$

The final gated representation is computed as:

$$\tilde{\mathbf{H}}_i = \alpha_i \odot \mathbf{H}_i^L + (1 - \alpha_i) \odot \mathbf{C}_i \tag{7}$$

Here, $\odot$ denotes element-wise multiplication, $\sigma(\cdot)$ is the sigmoid activation function, and Gate$(\cdot)$ is a shared learnable network. This formulation enables each agent to adjust its output by blending its own prediction with the global context, based on estimated confidence.

Global Voter To further integrate the confidence-adjusted outputs from all agents, we introduce a Global Voter that computes agent-wise collaboration weights $\mathbf{CW} \in \mathbb{R}^N$ based on the encoded input $\mathbf{X}_{\text{enc}}$ with MLP. These weights indicate the relative importance or contribution of each agent to final prediction. Each weight in $\mathbf{CW}$ is then broadcast to match the shape of its corresponding agent representation and used to perform a weighted sum over $\{\tilde{\mathbf{H}}_1^L, \tilde{\mathbf{H}}_2^L, \dots, \tilde{\mathbf{H}}_N^L\}$. This results in a unified latent representation $\mathbf{Z}$, which captures the aggregated knowledge across all agents.

Finally, $\mathbf{Z}$ is projected through a linear layer to produce the final multivariate time series forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{M \times L}$, where L denotes the forecast horizon.

5 Experiments

5.1 Experimental Setups

Datasets We evaluate our model on 11 real-world datasets covering different domains. In Electricity domain, we use ETTh1, ETTh2, ETTm1, and ETTm2 [77, 60]. The Environment domain includes Weather [60], PM2.5 [59], AQShunyi and AQWan [41] . The Nature domain consists of CzeLan, ZafNoo [41] as well as Temp [59]. For the ETT datasets, we adopt a 6:2:2 train/validation/test split, and a 7:1:2 split for the remaining datasets.

Implementation Details MAFS explores four distinct communication topologies for the agent interaction graph $\mathbf{G}$, including ring, star, chain, and fully-connected structures. Each forecasting agent is implemented using iTransformer architecture [28]. MAFS is trained in two stages: the first stage independently optimizes each agent with a learning rate of 1e-3 for 10 epochs; the second stage freezes agent parameters and finetunes only the topology-aware weight adjuster and agent-rated voting aggregator for another 10 epochs with a reduced learning rate of 1e-4. We set a hidden dimension of 128, and configure the encoder with 2 layer for all datasets. The number of agents

Table 1: Comparison of long-term time series forecasting methods on 11 datasets using MSE and MAE (lower is better). Best results are marked in **red** ; second-best results are underlined in blue .

Methods	MAFS (Ours)		iTransformer [2024]		TimeMixer [2024]		PatchTST [2024]		Crossformer [2023]		TiDE [2024]		TimesNet [2023]		DLinear [2023]		Autoformer [2021]		Informer [2021]	
Datasets	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.433**	**0.437**	0.467	0.466	0.447	0.44	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.45	0.456	0.452	0.496	0.487	1.04	0.795
ETTh2	**0.356**	**0.394**	0.386	0.415	0.364	0.395	0.387	0.407	0.942	0.684	0.611	0.55	0.414	0.427	0.559	0.515	0.45	0.459	4.431	1.729
ETTm1	**0.366**	**0.388**	0.383	0.403	0.381	0.395	0.387	0.4	0.513	0.496	0.419	0.419	0.4	0.406	0.403	0.407	0.588	0.517	0.961	0.734
ETTm2	**0.265**	**0.321**	0.29	0.339	0.275	0.323	0.281	0.326	0.757	0.61	0.358	0.404	0.291	0.333	0.35	0.401	0.327	0.371	1.41	0.81
Weather	**0.233**	**0.267**	0.243	0.276	0.24	0.271	0.259	0.281	0.259	0.315	0.271	0.32	0.259	0.287	0.265	0.317	0.338	0.382	0.634	0.548
AQShunyi	0.701	0.509	0.723	0.515	0.719	0.529	0.705	0.509	**0.694**	**0.504**	0.778	0.554	0.726	0.516	0.706	0.522	0.764	0.541	0.782	0.545
AQWan	0.802	0.503	0.817	0.507	0.828	0.499	0.812	0.499	**0.786**	**0.49**	0.856	0.536	0.813	0.5	0.818	0.512	0.84	0.525	0.866	0.525
CzeLan	**0.222**	**0.271**	0.232	0.28	0.228	0.28	0.227	0.29	0.956	0.576	0.237	0.303	0.224	0.285	0.284	0.342	0.307	0.355	0.316	0.355
ZafNoo	0.52	0.451	0.541	0.468	0.538	**0.44**	0.511	0.465	**0.494**	0.455	0.569	0.498	0.537	0.465	0.496	0.451	0.725	0.599	0.744	0.602
PM2.5	**0.398**	**0.414**	0.421	0.421	0.415	0.436	0.46	0.479	0.456	0.472	0.481	0.497	0.473	0.492	0.453	0.477	0.515	0.524	0.539	0.561
Temp	**0.14**	**0.288**	0.173	0.321	0.146	0.304	0.147	0.306	0.206	0.423	0.164	0.338	0.208	0.428	0.159	0.327	0.244	0.5	0.238	0.492

N is selected from the range $\{4, 8, 12, 16, 20, 24\}$ to investigate the impact of varying agent scale. All experiments adopt a symmetric prediction setting, where the input sequence length equals the forecasting horizon [76]. More Details are available at Appendix C.

Baselines The proposed method is evaluated against a range of representative baselines, which can be categorized by model architecture. Transformer-based models include Informer [77], Autoformer [60], Crossformer [76], PatchTST [38], and iTransformer [28]. Linear-based models such as TimeMixer [58], TiDE [6] and DLinear [73] focus on efficient feature extraction and forecsating. Periodicity-based models, such as TimesNet [61], enhance forecasting performance by modeling multi-period temporal patterns.

5.2 Main Results

The experimental results are presented in Table 1, where we comprehensively evaluate the proposed MAFS against a range of SOTA models on long-term time series forecasting benchmarks. Experimental results consistently demonstrate the superior performance and robustness of MAFS across diverse datasets and prediction horizons. Specifically, MAFS achieves an average improvement of 3.24% in MSE and 1.71% in MAE over the recent SOTA model TimeMixer, confirming the effectiveness of our multi-agent collaborative framework. Furthermore, although MAFS adopts the iTransformer backbone, which is not the most competitive SOTA model in terms of forecasting accuracy, our multi-agent framework effectively overcomes this limitation. By leveraging agent specialization and structured inter-agent collaboration, MAFS achieves an average improvement of 6.35% in MSE and 4.03% in MAE compared to iTransformer, successfully reaching state-of-the-art performance across multiple benchmarks. In addition, MAFS ranks first on 16 out of 22 evaluation metrics and secures a top-2 ranking in 20 out of 22 metrics (covering both MSE and MAE across 11 datasets), showcasing its strong generalization capability across various application domains. In summary, MAFS breaks the limitations of monolithic forecasting models by introducing a flexible and adaptive multi-agent collaboration mechanism, delivering more accurate and reliable predictions across diverse and challenging time series forecasting scenarios. Full results are available at Appendix G.

5.3 Ablation Analysis

To evaluate the effectiveness of each key component in our proposed MAFS framework, we conduct a series of ablation studies. The experimental results are presented in Table 2. We evaluate the removal of three critical modules, **1) w/o Comm**: Disables inter-agent communication by removing the shared semantic vector h_c, preventing collaborative reasoning. **2) w/o AVA**: Removes the Agent-Rated Voting Aggregator, directly averaging agent embeddings without adaptive gating and collaboration weights. **3) w/o STS**: Disables Sub-task Specialization, assigning identical forecasting tasks to all agents instead of specialized sub-tasks.

Main results. (1) w/o Comm: Without communication, MSE and MAE increase by 4.18% and 2.62%, highlighting its role in enabling agents to share complementary temporal information. (2) Removing AVA leads to a significant MSE and MAE increase of 7.24% and 5.70%, confirming its importance in adaptively integrating agent outputs for accurate forecasting. (3) Without sub-task

Table 2: Ablation Study on the Contribution of Communication, Aggregation, and Specialization Modules in MAFS. **Bold** values indicate the best performance.

Variant	Metric	ETTh1	ETTh2	ETTm1	ETTm2	Weather	AQShunyi	AQWan	CzeLan	ZafNoo	PM2.5	Temp
MAFS	MSE	**0.433**	**0.356**	**0.366**	**0.265**	**0.233**	**0.701**	**0.802**	**0.222**	**0.520**	**0.398**	**0.140**
	MAE	**0.437**	**0.394**	**0.388**	**0.321**	**0.267**	**0.509**	**0.503**	**0.271**	**0.451**	**0.414**	**0.288**
w/o Comm	MSE	0.445	0.364	0.375	0.277	0.238	0.719	0.824	0.228	0.529	0.439	0.167
	MAE	0.446	0.402	0.398	0.335	0.273	0.517	0.513	0.279	0.456	0.427	0.312
w/o AVA	MSE	0.454	0.373	0.379	0.288	0.265	0.722	0.845	0.253	0.536	0.421	0.158
	MAE	0.456	0.409	0.403	0.335	0.293	0.520	0.532	0.292	0.487	0.426	0.301
w/o STS	MSE	0.461	0.384	0.374	0.286	0.244	0.726	0.823	0.243	0.546	0.417	0.171
	MAE	0.462	0.412	0.397	0.332	0.275	0.517	0.512	0.285	0.475	0.415	0.317

specialization, MSE and MAE rise by 6.97% and 3.93%, demonstrating that specialized agents improve representation and capture diverse signal characteristics.

5.4 Evaluating the Advantage of MAFS over Single Models

As shown in Figure 4, MAFS consistently outperforms the single-agent model across 11 datasets, achieving an average improvement of 6.35% in MSE and 4.03% in MAE, demonstrating the effectiveness of collaborative forecasting. The most significant improvement occurs on the Temp dataset, with a 19.08% reduction in MSE and 10.28% in MAE. This gain likely results from strong temporal patterns and variable dependencies in temperature data, which are better captured by specialized agents and collaborative reasoning. Overall, these results confirm that MAFS effectively realizes collective intelligence, enabling specialized agents to achieve superior forecasting accuracy, and highlighting the advantages of a multi-agent framework over monolithic models.

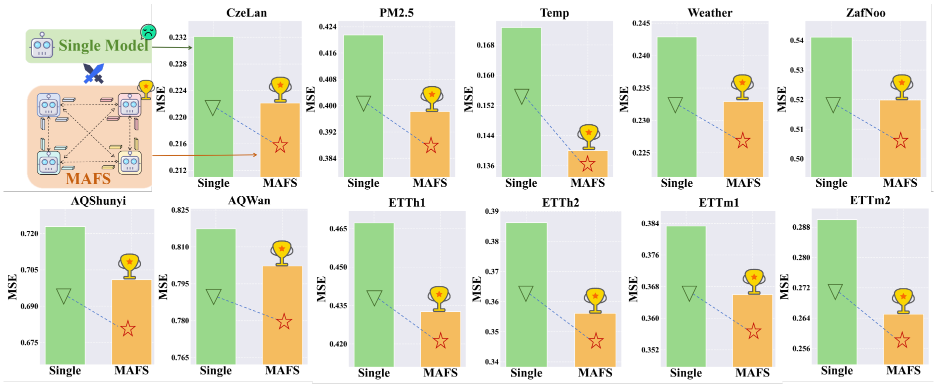

Figure 4: Performance comparison between single forecasting model and MAS forecasting system

5.5 Analysis of MAFS Scaling and Communication Structures

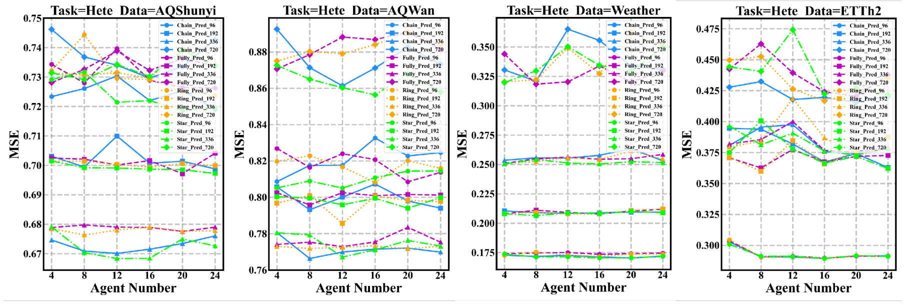

Figure 5: Performance comparison under varying agent numbers and communication structures

The results are shown in Figure 5. (1) Agent Scaling: Increasing the number of agents generally improves performance by enhancing modeling diversity and specialization. However, beyond 16 agents, improvements become marginal, indicating a saturation point. Minor performance fluctuations are trivial and likely caused by random initialization or local optima. Full results are available at Appendix E. (2) Communication Structures: Among chain, ring, fully-connected, and star topologies, the star structure consistently yields better and more stable results. Its centralized design effectively integrates global information, reducing noise amplification and ensuring critical trends are shared across agents. In conclusion, using 16 agents with a star communication structure offers the best trade-off between performance and complexity for time series forecasting.

5.6 Impact of Sub-task Division on Agent Specialization

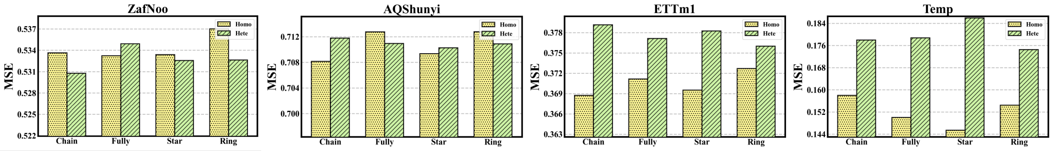

Figure 6: Comparison of agent specialization under different sub-task division strategies

Figure 6 shows the average forecasting accuracy under different sub-task divisions across prediction horizons. First, in most cases, the performance difference between homogeneous and heterogeneous task divisions is minor. For example, the performance gap remains within 1% between ZafNoo and AQShunyi, suggesting that task division strategies have limited influence under such data conditions. Second, for datasets with smaller variance and more stable patterns (e.g., ETTm1 and Temp), homogeneous task division significantly outperforms heterogeneous division. This indicates that consistent modeling strategies are better suited for stable datasets without introducing unnecessary task diversity. In summary, the choice of sub-task division should consider the characteristics of the target dataset, with homogeneous division preferred for stable data and heterogeneous division applicable when greater diversity is required. Full results are available at Appendix D.

5.7 Case Study

As shown in Figure 7, we take ETTm2 to illustrate effectiveness of MAFS through sub-task evaluation, voting scores of different agents, and the learned communication weights. (1) Each agent can successfully specialize and perform well on its assigned sub-task. (2) The voting results show that all agents actively contribute to the final decision, avoiding issues such as agent collapse or over-reliance on a single expert. (3) In the second stage, the learned adaptive communication weights reflect task-aware information exchange, optimizing collaboration between agents under a fixed topology.

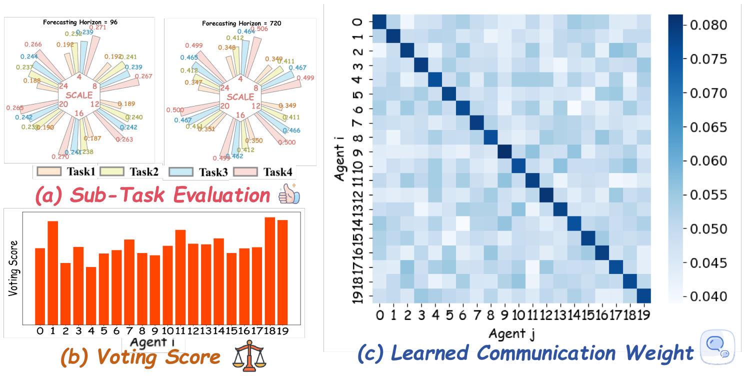

Figure 7: Case study of MAFS on ETTm2.

6 Conclusion

In this work, we propose MAFS, a novel multi-agent forecasting system that introduces collective intelligence into time series forecasting through principled task decomposition, structured inter-agent communication, and a two-stage voting aggregator. Extensive experiments on 11 real-world datasets demonstrate that MAFS consistently outperforms state-of-the-art baselines, achieving significant improvements in both accuracy and robustness. These results highlight the effectiveness of collaborative forecasting and offer a new perspective for addressing the challenges of complex and evolving temporal patterns. This work can open new directions for modular, adaptive, and interpretable time series forecasting.

7 Acknowledgement

This paper is partially supported by the National Natural Science Foundation of China (No.62502488, No.12227901), Natural Science Foundation of Jiangsu Province (BK20240460), the grant from State Key Laboratory of Resources and Environmental Information System. The AI-driven experiments, simulations and model training were performed on the robotic AI-Scientist platform of Chinese Academy of Sciences.

References

[1] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 5th edition, 2015.

[2] Lu Chen, Qilu Zhong, Xiaokui Xiao, Yunjun Gao, Pengfei Jin, and Christian S Jensen. Price-and-time-aware dynamic ridesharing. In *2018 IEEE 34th international conference on data engineering (ICDE)*, pages 1061–1072. IEEE, 2018.

[3] Lu Chen, Yunjun Gao, Ziquan Fang, Xiaoye Miao, Christian S Jensen, and Chenjuan Guo. Real-time distributed co-movement pattern detection on streaming trajectories. *Proceedings of the VLDB Endowment*, 12(10):1208–1220, 2019.

[4] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

[5] Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. *arXiv preprint arXiv:2410.08115*, 2024.

[6] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan K Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *Transactions on Machine Learning Research*, 2024.

[7] Xin Ding, Lu Chen, Yunjun Gao, Christian S Jensen, and Hujun Bao. Ultraman: A unified platform for big trajectory data management and analytics. *Proceedings of the VLDB Endowment*, 11(7):787–799, 2018.

[8] Shanghua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37:140589–140631, 2024.

[9] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[10] Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12608–12616, 2024.

[11] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Qihe Huang, Zhengyang Zhou, Kuo Yang, Gengyu Lin, Zhongchao Yi, and Yang Wang. Leret: Language-empowered retentive network for time series forecasting. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 4165–4173. International Joint Conferences on Artificial Intelligence Organization, 8

2024. doi: 10.24963/ijcai.2024/460. URL https://doi.org/10.24963/ijcai.2024/460. Main Track.

[13] Qihe Huang, Zhengyang Zhou, Kuo Yang, and Yang Wang. Exploiting language power for time series forecasting with exogenous variables. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 4043–4052, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714793. URL https://doi.org/10.1145/3696410.3714793.

[14] Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

[15] 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. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.

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

[18] David Kinny and Michael Georgeff. Modelling and design of multi-agent systems. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 1–20. Springer, 1996.

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1609.02907.

[20] Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. *Advances in Neural Information Processing Systems*, 37:131224–131246, 2024.

[21] Xinyu Li, Yuchen Luo, Hao Wang, Haoxuan Li, Liuhua Peng, Feng Liu, Yandong Guo, Kun Zhang, and Mingming Gong. Towards accurate time series forecasting via implicit decoding. *Advances in Neural Information Processing Systems*, 2025.

[22] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[23] Shengsheng Lin, Weiwei Lin, Wentai Wu, Songbo Wang, and Yongxiang Wang. Petformer: Long-term time series forecasting via placeholder-enhanced transformer. *arXiv*, 2023.

[24] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023.

[25] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.

[26] 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*, 2024.

[27] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, 2022.

[28] 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*, 2024.

[29] Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024.

[31] Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

[32] Jiaming Ma, Zhiqing Cui, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Zhe Zhao, and Yang Wang. Causal learning meet covariates: Empowering lightweight and effective nationwide air quality forecasting. 2025.

[33] Jiaming Ma, Binwu Wang, Qihe Huang, Guanjun Wang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. Mofo: Empowering long-term time series forecasting with periodic pattern modeling. In *Advances in Neural Information Processing Systems*, 2025.

[34] Jiaming Ma, Binwu Wang, Guanjun Wang, Kuo Yang, Zhengyang Zhou, Pengkun Wang, Xu Wang, and Yang Wang. Less but more: Linear adaptive graph learning empowering spatiotemporal forecasting. In *Advances in Neural Information Processing Systems*, 2025.

[35] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Bist: A lightweight and efficient bi-directional model for spatiotemporal prediction. *Proceedings of the VLDB Endowment*, 18(6):1663–1676, 2025.

[36] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Xu Wang, and Yang Wang. Robust spatio-temporal centralized interaction for ood learning. In *Forty-second International Conference on Machine Learning*, 2025.

[37] Jiaming Ma, Binwu Wang, Pengkun Wang, Zhengyang Zhou, Yudong Zhang, Xu Wang, and Yang Wang. Mobimixer: A multi-scale spatiotemporal mixing model for mobile traffic prediction. *IEEE Transactions on Mobile Computing*, 2025.

[38] 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*, 2023.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[40] Tingrui Qiao, Caroline Walker, Chris Cunningham, and Yun Sing Koh. Thematic-lm: A llm-based multi-agent system for large-scale thematic analysis. In *Proceedings of the ACM on Web Conference 2025*, pages 649–658, 2025.

[41] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. In *Proc. VLDB Endow.*, pages 2363–2377, 2024.

[42] Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. Tab: Unified benchmarking of time series anomaly detection methods. In *Proc. VLDB Endow.*, pages 2775–2789, 2025.

[43] Xiangfei Qiu, Xingjian Wu, Hanyin Cheng, Xvyuan Liu, Chenjuan Guo, Jilin Hu, and Bin Yang. DBLoss: Decomposition-based loss function for time series forecasting. In *NeurIPS*, 2025.

[44] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *SIGKDD*, pages 1185–1196, 2025.

[45] 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. In *International Conference on Learning Representations*, 2017.

[46] 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. *arXiv preprint arXiv:2409.16040*, 2024.

[47] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

[48] Wiebe Van der Hoek and Michael Wooldridge. Multi-agent systems. *Foundations of Artificial Intelligence*, 3:887–928, 2008.

[49] Binwu Wang, Yudong Zhang, Xu Wang, Pengkun Wang, Zhengyang Zhou, Lei Bai, and Yang Wang. Pattern expansion and consolidation on evolving graphs for continual traffic prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2223–2232, 2023.

[50] Hao Wang, Zhiyu Wang, Yunlong Niu, Zhaoran Liu, Haozhe Li, Yilin Liao, Yuxin Huang, and Xinggao Liu. An accurate and interpretable framework for trustworthy process monitoring. *IEEE Transactions on Artificial Intelligence*, 5(5):2241–2252, 2023.

[51] Hao Wang, Haoxuan Li, Xu Chen, Mingming Gong, Zhichao Chen, et al. Optimal transport for time series imputation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] Hao Wang, Lichen Pan, Yuan Shen, Zhichao Chen, Degui Yang, Yifei Yang, Sen Zhang, Xinggao Liu, Haoxuan Li, and Dacheng Tao. Fredf: Learning to forecast in the frequency domain. In *The Thirteenth International Conference on Learning Representations*, 2025.

[53] Hao Wang, Licheng Pan, Zhichao Chen, Xu Chen, Qingyang Dai, Lei Wang, Haoxuan Li, and Zhouchen Lin. Time-o1: Time-series forecasting needs transformed label alignment. *Advances in Neural Information Processing Systems*, 2025.

[54] Haotian Wang, Haoxuan Li, Hao Zou, Haoang Chi, Long Lan, Wanrong Huang, and Wenjing Yang. Effective and efficient time-varying counterfactual prediction with state-space models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[55] 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 *The Eleventh International Conference on Learning Representations*, 2023.

[56] Lei Wang, Shanshan Huang, Chunyuan Zheng, Jun Liao, Xiaofei Zhu, Haoxuan Li, and Li Liu. Mitigating data imbalance in time series classification based on counterfactual minority samples augmentation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2962–2973, 2025.

[57] Shilong Wang, Guibin Zhang, Miao Yu, Guancheng Wan, Fanci Meng, Chongye Guo, Kun Wang, and Yang Wang. G-safeguard: A topology-guided security lens and treatment on llm-based multi-agent systems. *arXiv preprint arXiv:2502.11127*, 2025.

[58] 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. In *The Twelfth International Conference on Learning Representations*, 2024.

[59] Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, pages 163–166, 2020.

[60] 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.

[61] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

[62] Xingjian Wu, Xiangfei Qiu, Hanyin Cheng, Zhengyu Li, Jilin Hu, Chenjuan Guo, and Bin Yang. Enhancing time series forecasting through selective representation spaces: A patch perspective. In *NeurIPS*, 2025.

[63] Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A koopman-kalman enhanced variational autoencoder for probabilistic time series forecasting. In *ICML*, 2025.

[64] Xingjian Wu, Xiangfei Qiu, Zhengyu Li, Yihang Wang, Jilin Hu, Chenjuan Guo, Hui Xiong, and Bin Yang. CATCH: Channel-aware multivariate time series anomaly detection via frequency patching. In *ICLR*, 2025.

[65] Zhao Xinjie, Fan Gao, Rui Yang, Yingjian Chen, Yuyang Wang, Ying Zhu, Jiacheng Tang, and Irene Li. Reagent: Reversible multi-agent reasoning for knowledge-enhanced multi-hop qa. *arXiv preprint arXiv:2503.06951*, 2025.

[66] Vladyslav Yevsieiev. *Using Multi-Agent Systems in the Management of Collaborative Robots*. PhD thesis, 2025.

[67] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045, 2024.

[68] Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, et al. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*, 2024.

[69] Wenzhen Yue, Xianghua Ying, Ruohao Guo, DongDong Chen, Ji Shi, Bowei Xing, Yuqing Zhu, and Taiyan Chen. Sub-adjacent transformer: Improving time series anomaly detection with reconstruction error from sub-adjacent neighborhoods. *arXiv preprint arXiv:2404.18948*, 2024.

[70] Wenzhen Yue, Yong Liu, Hao Wang, Haoxuan Li, Xianghua Ying, Ruohao Guo, Bowei Xing, and Ji Shi. Olinear: A linear model for time series forecasting in orthogonally transformed domain. *Advances in Neural Information Processing Systems*, 2025.

[71] Wenzhen Yue, Yong Liu, Xianghua Ying, Bowei Xing, Ruohao Guo, and Ji Shi. Freeformer: Frequency enhanced transformer for multivariate time series forecasting. *arXiv preprint arXiv:2501.13989*, 2025.

[72] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. *arXiv preprint arXiv:2309.05444*, 2023.

[73] 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.

[74] Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025.

[75] Yao Zhang, Zijian Ma, Yunpu Ma, Zhen Han, Yu Wu, and Volker Tresp. Webpilot: A versatile and autonomous multi-agent system for web task execution with strategic exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23378–23386, 2025.

[76] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

[77] 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 *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[78] Zhengyang Zhou, Qihe Huang, Gengyu Lin, Kuo Yang, LEI BAI, and Yang Wang. GReto: Remedying dynamic graph topology-task discordance via target homophily. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8duT3mi_5n.

[79] Zhengyang Zhou, Qihe Huang, Binwu Wang, Jianpeng Hou, Kuo Yang, Yuxuan Liang, Yu Zheng, and Yang Wang. Coms2t: A complementary spatiotemporal learning system for data-adaptive model evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47 (10):8506–8523, 2025. doi: 10.1109/TPAMI.2025.3576805.

Appendix

A Details of Forecasting Sub-task Decomposition

Homogeneous sub-task Decomposition: Multi-Scale Temporal Forecasting. Under the homogeneous setting, each agent is tasked with predicting future values at a distinct temporal resolution. Specifically, the i-th agent forecasts a future subsequence of length $(L/K) \cdot i$, where L is the full prediction horizon and K is the number of agents. Formally, the prediction from agent i is denoted as

$$\hat{\mathbf{O}}_i = \mathbf{Agent}_i(\mathbf{X}), \quad \hat{\mathbf{O}}_i \in \mathbb{R}^{(L/K)\cdot i \times M},$$

where $\mathbf{X} \in \mathbb{R}^{T \times M}$ is the multivariate historical sequence of length T and M is the number of variables. This design ensures that each agent captures forecasting dynamics at a specific temporal scale, ranging from short-term fluctuations to long-term evolution, thereby forming a multi-resolution ensemble.

Heterogeneous sub-task Decomposition: Multi-Aspect Signal Forecasting. To capture different latent components of the future signal, we further design a heterogeneous sub-task scheme, where each agent is specialized on a predefined signal property. This enables orthogonal learning objectives across agents, contributing complementary views to the ensemble. We define four types of forecasting sub-tasks in this setting:

We first define the future sequence as $\mathbf{Y} \in \mathbb{R}^{L \times M}$, where L is the prediction horizon. Let each agent process $\mathbf{Y}$ to extract distinct forecasting targets.

(1) Trend forecasting. We estimate the low-frequency trend component by applying temporal average pooling over a padded version of $\mathbf{Y}$:

$$\mathbf{T} = \mathbf{AvgPool}(\mathbf{Pad}(\mathbf{Y})) \in \mathbb{R}^{T \times M},$$

where $\mathbf{Pad}(\cdot)$ aligns the length of $\mathbf{Y}$ with the pooling window and $\mathbf{AvgPool}(\cdot)$ computes segment-wise means. This sub-task enables the agent to focus on slowly evolving components in the signal.

(2) Seasonality forecasting. Seasonal dynamics are extracted by removing the estimated trend from the original signal:

$$\mathbf{S} = \mathbf{Y} - \mathbf{F}, \quad \text{where} \quad \mathbf{F} = \mathbf{TrendModel}(\mathbf{Y}),$$

with $\mathbf{F}$ being a smoothed version of $\mathbf{Y}$, either computed or learned. This task guides the agent to attend to periodic or residual structures.

(3) Spectral energy forecasting. To encode frequency-domain characteristics, we apply the real-valued Fast Fourier Transform (FFT) and take the magnitude:

$$\mathbf{E} = |\mathbf{FFT}(\mathbf{Y})| \in \mathbb{R}^{(L//2+1)\times M}.$$

The resulting spectral energy profile provides insights into dominant frequencies and periodicities in the future signal.

(4) Statistical descriptor forecasting. We summarize $\mathbf{Y}$ using a fixed set of descriptive statistics, capturing distributional properties as follows:

$$\mathbf{D} = [\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\kappa}, \mathbf{y}_{\max}, \mathbf{y}_{\min}] \in \mathbb{R}^{6\times M},$$

where

$$\boldsymbol{\mu} = \mathbf{Mean}(\mathbf{Y}), \quad \boldsymbol{\sigma} = \mathbf{Std}(\mathbf{Y}), \quad \boldsymbol{\gamma} = \mathbf{Skewness}(\mathbf{Y}), \quad \boldsymbol{\kappa} = \mathbf{Kurtosis}(\mathbf{Y}),$$

and

$$\mathbf{y}_{\max} = \max_t \mathbf{Y}[t], \quad \mathbf{y}_{\min} = \min_t \mathbf{Y}[t].$$

This task allows the agent to model global signal characteristics in a compact form.

B More Dataset Details

Table 3: Summary of the 11 datasets used in our experiments, covering diverse domains, temporal resolutions, and feature dimensions.

Dataset	Variate	Input Length	Predict Length	Information	Frequency	Split
ETTh1	7	$96 \sim 720$	$96 \sim 720$	Electricity	15mins	6:2:2
ETTh2	7	$96 \sim 720$	$96 \sim 720$	Electricity	15mins	6:2:2
ETTm1	7	$96 \sim 720$	$96 \sim 720$	Electricity	15mins	6:2:2
ETTm2	7	$96 \sim 720$	$96 \sim 720$	Electricity	15mins	6:2:2
Weather	21	$96 \sim 720$	$96 \sim 720$	Environment	10mins	7:1:2
Temperature	108	$96 \sim 720$	$96 \sim 720$	Environment	3hours	7:1:2
AOShunyi	11	$96 \sim 720$	$96 \sim 720$	Environment	1hour	7:1:2
AQWan	11	$96 \sim 720$	$96 \sim 720$	Environment	1hour	7:1:2
PM2.5	108	$96 \sim 720$	$96 \sim 720$	Nature	3hours	7:1:2
ZafNoo	11	$96 \sim 720$	$96 \sim 720$	Nature	30mins	7:1:2
CzeLan	11	$96 \sim 720$	$96 \sim 720$	Nature	30mins	7:1:2

As shown in Table 3, we evaluate our framework on 11 multivariate time series datasets spanning three real-world application domains: Electricity, Environment, and Nature. These datasets exhibit diverse characteristics in terms of feature dimensionality, sampling frequency, and domain semantics, offering a comprehensive benchmark for assessing forecasting performance under different temporal patterns and data complexities. The Electricity domain includes four ETT datasets: ETTh1, ETTh2, ETTm1, and ETTm2 [77, 60]. Each contains 7 variates recorded at 15-minute intervals, primarily reflecting industrial power consumption dynamics. The Environment domain covers Weather [60], Temperature [59], AQShunyi, and AQWan [41]. Weather contains 21 meteorological variables sampled every 10 minutes, while Temperature and PM2.5 have higher dimensionality (108 variates) at a coarser 3-hour frequency. AQShunyi and AQWan offer 11-dimensional hourly air quality readings from different regions in Beijing. The Nature domain consists of PM2.5 [59], ZafNoo, and

CzeLan [41], all of which represent long-range environmental trends collected over various regions, with sampling frequencies ranging from 30 minutes to 3 hours. Across all datasets, the input and prediction sequence lengths are consistently selected from the range 96 to 720 to ensure uniform temporal coverage across tasks. For the ETT datasets, we follow prior work and adopt a 6:2:2 split for training, validation, and testing. For the remaining datasets, we use a 7:1:2 split. This ensures both consistent evaluation and fair comparison with previous studies.

C More Implementation Details

All experiments are conducted on a server equipped with 8 NVIDIA A100 GPUs (80GB memory each). MAFS is implemented using PyTorch 1.13.0 [39] and optimized using the Adam optimizer [17] with an L2 loss. We investigate four distinct communication topologies for the agent interaction graph $\mathbf{G}$, including ring, star, chain, and fully-connected structures. Each forecasting agent is instantiated using the iTransformer architecture [28], with a unified configuration across all experiments: a fixed learning rate of 1e-3, a hidden dimension of 128, and 2 encoder layers. To evaluate the scalability of MAFS, we vary the number of agents $N \in \{4, 8, 12, 16, 20, 24\}$. MAFS training is conducted in two stages. In the first stage, each agent is independently optimized on its assigned sub-task using a learning rate of 1e-3 for 10 epochs. In the second stage, we freeze the parameters of all forecasting agents and jointly finetune only the topology-aware weight adjuster and the agent-rated voting aggregator. This stage also runs for 10 epochs, using a smaller learning rate of 1e-4 to enable stable convergence during topology adaptation and collaborative forecasting. For all tasks, we adopt a symmetric prediction setting where the historical input length equals the forecasting horizon, following the protocol in [76], to enable consistent evaluation across datasets. To ensure fair comparison, we re-run iTransformer under our unified setting (including learning rate, hidden size, and number of layers). For all baselines other than iTransformer, we use reported results from the iTransformer and TFB [41] papers, except for the `temp` and `pm2.5` datasets, where we re-run all baseline models. Importantly, we identified and corrected a bug in the original evaluation process: the `test_loader` was configured with `drop_last=True` during testing, which led to the exclusion of final test batches. We explicitly set `drop_last=False` to ensure fair and complete evaluation [41].

D Full Results of Comparison Between Two Types of Sub-tasks Division

Figure 8 to Figure 18 present a comprehensive comparison of homogeneous-correlated and heterogeneous-correlated sub-task configurations across all 11 datasets. The results demonstrate that each configuration exhibits distinct advantages under different data characteristics, emphasizing the need for adaptive task decomposition in multi-agent forecasting.

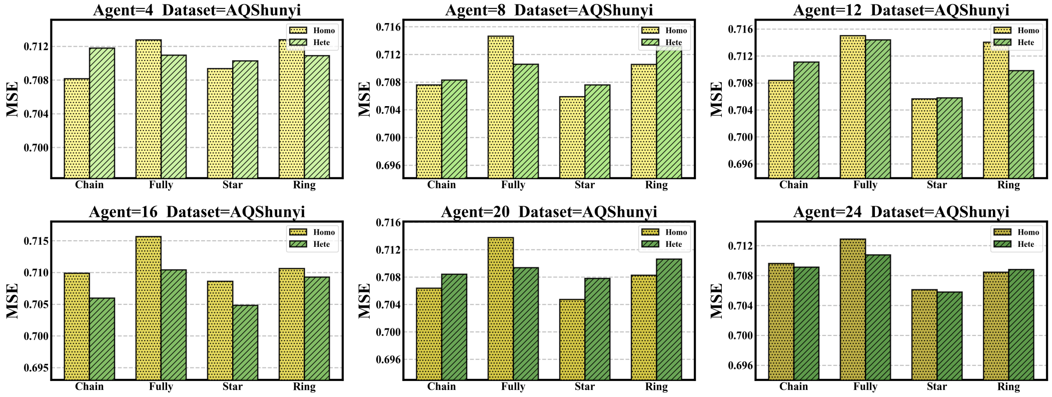

Figure 8: Performance comparison of homogeneous-correlated and heterogeneous-correlated sub-tasks on AQShunyi.

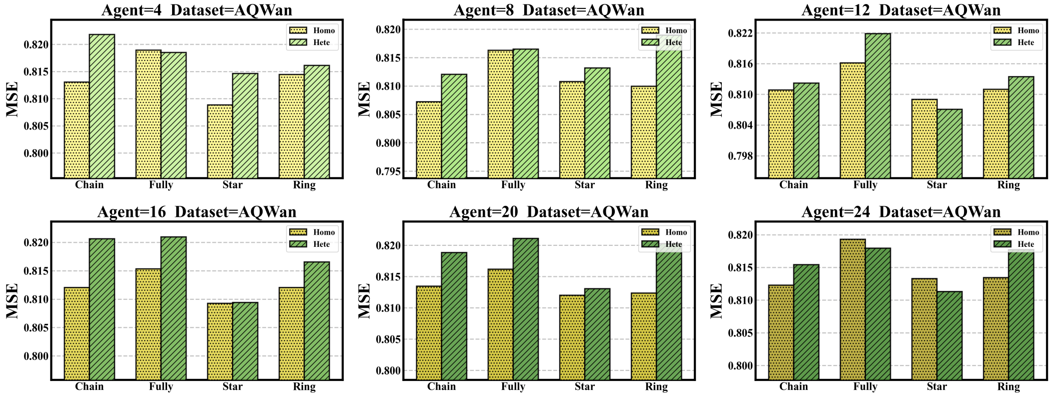

Figure 9: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on AQWan.

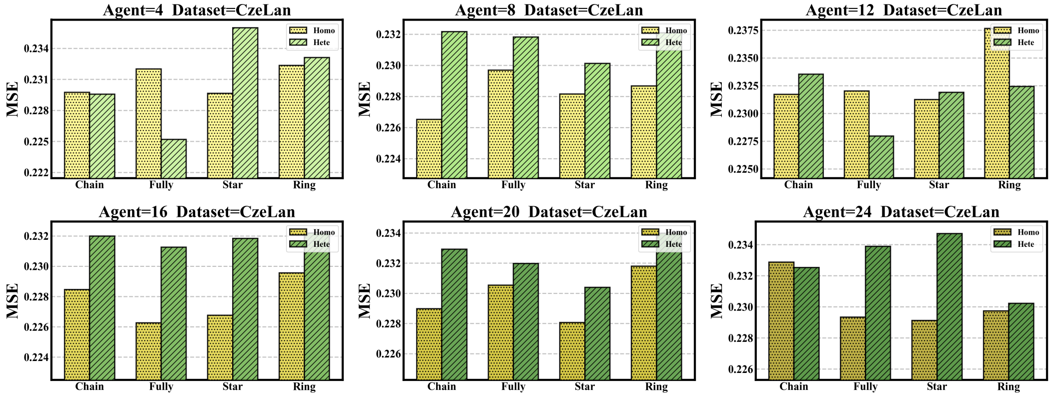

Figure 10: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on CzeLan.

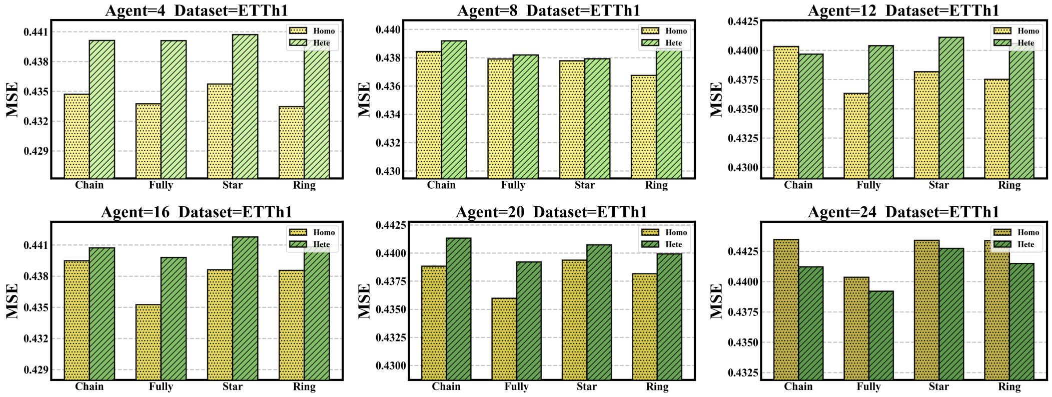

Figure 11: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on ETTh1.

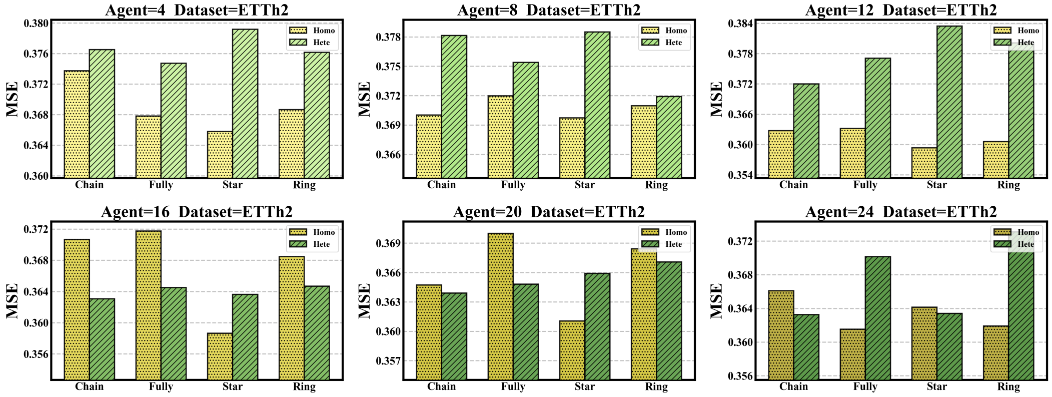

Figure 12: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on ETTh2.

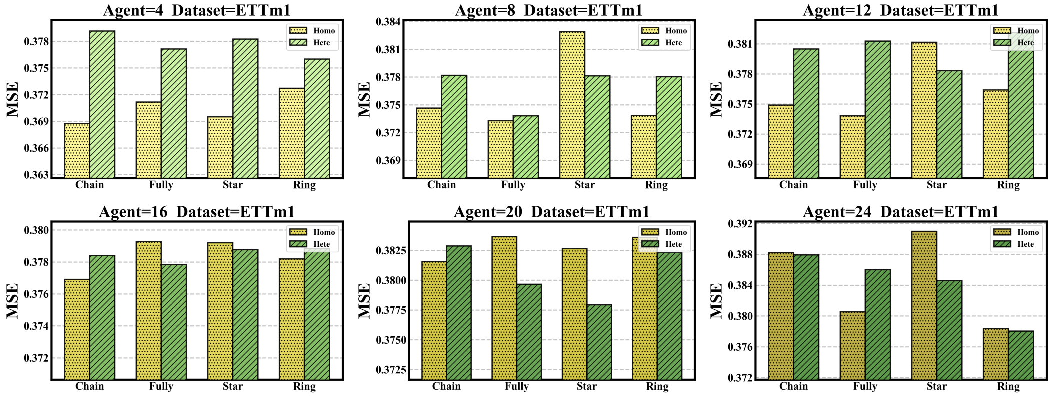

Figure 13: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on ETTm1.

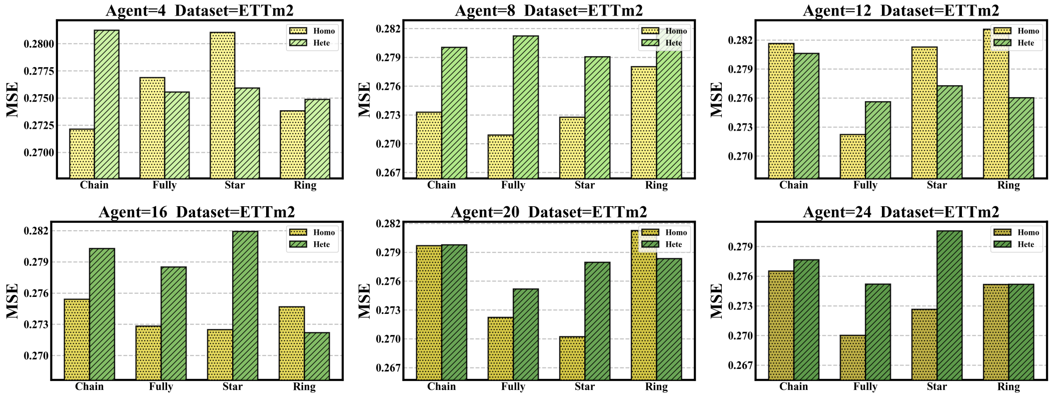

Figure 14: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on ETTm2.

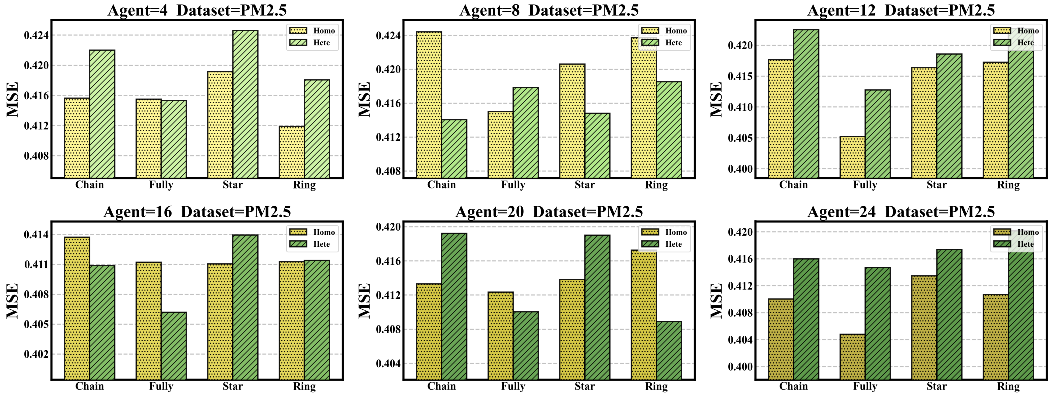

Figure 15: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on pm2.5.

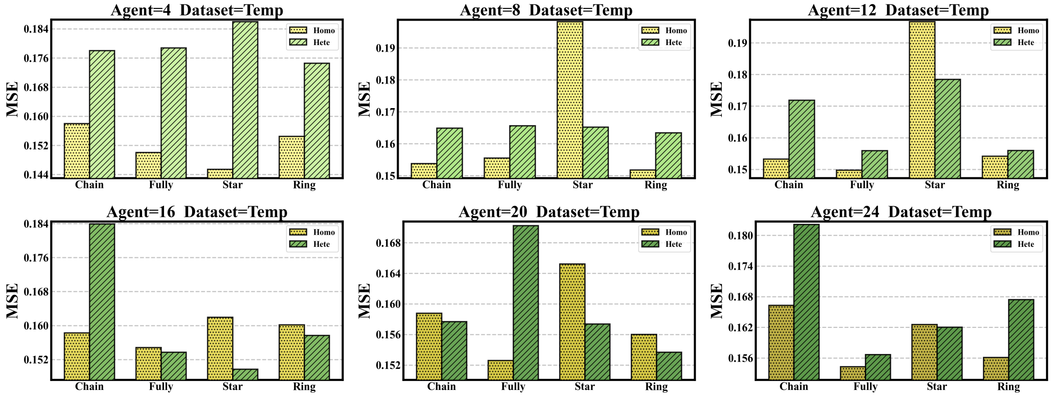

Figure 16: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on temp.

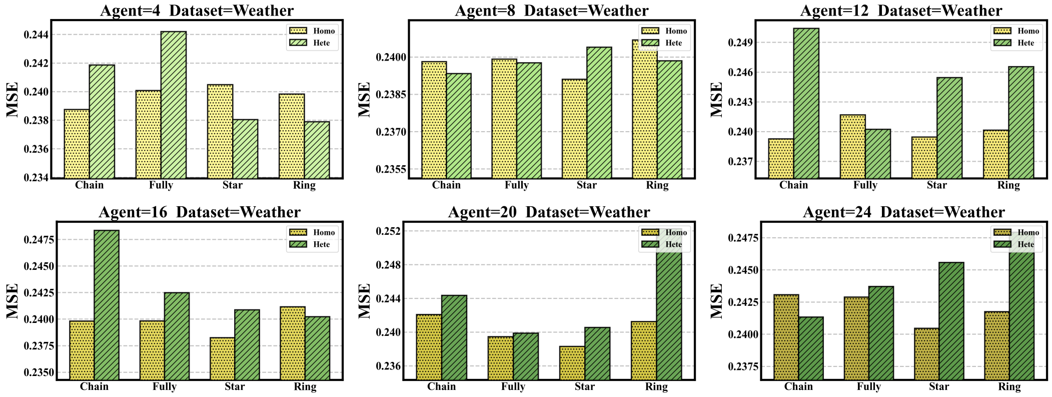

Figure 17: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on weather.

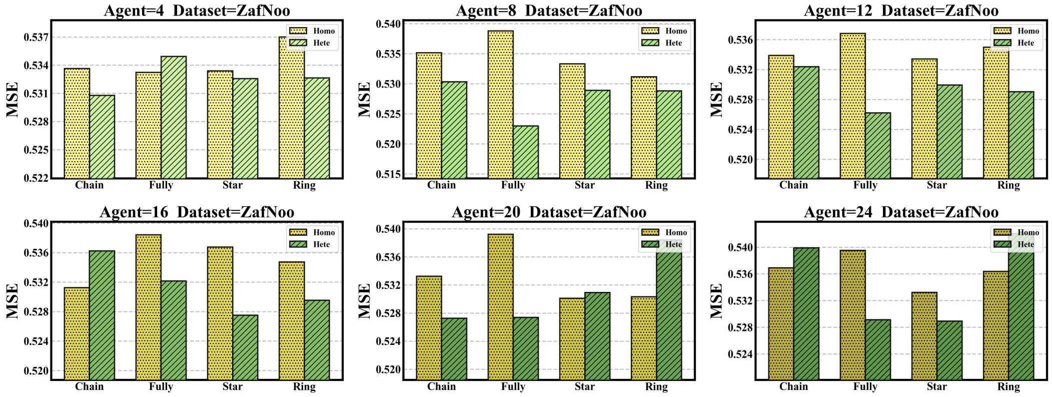

Figure 18: Performance comparison of homogeneous-correlated and heterogeneous-correlated subtasks on ZafNoo.

E Full Results of Different MAFS Scales

Figure 19 to Figure 29 show the full results of MAFS under varying numbers of agents across all datasets. Overall, we observe that increasing the number of agents generally leads to more stable forecasting performance, with reduced variance across different runs. Although more agents do not always yield the best accuracy, the ensemble effect tends to enhance robustness. This finding supports the scalability of MAFS in diverse time series scenarios and highlights the trade-off between performance and computational overhead when scaling agent populations.

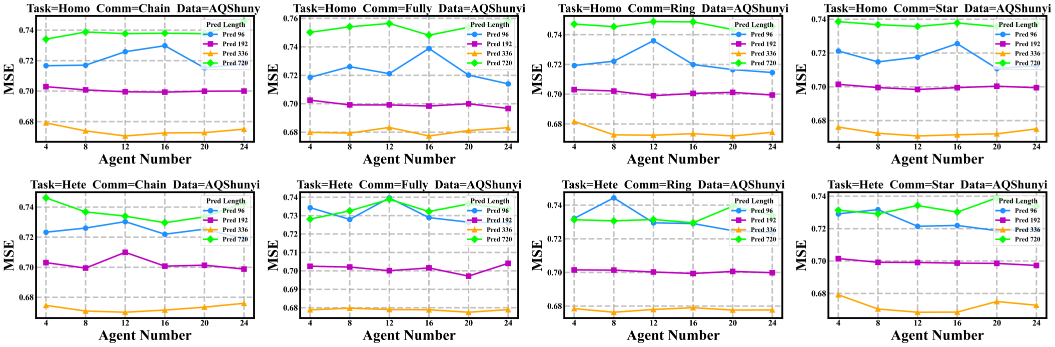

Figure 19: Performance comparison of different agent numbers on AQShunyi.

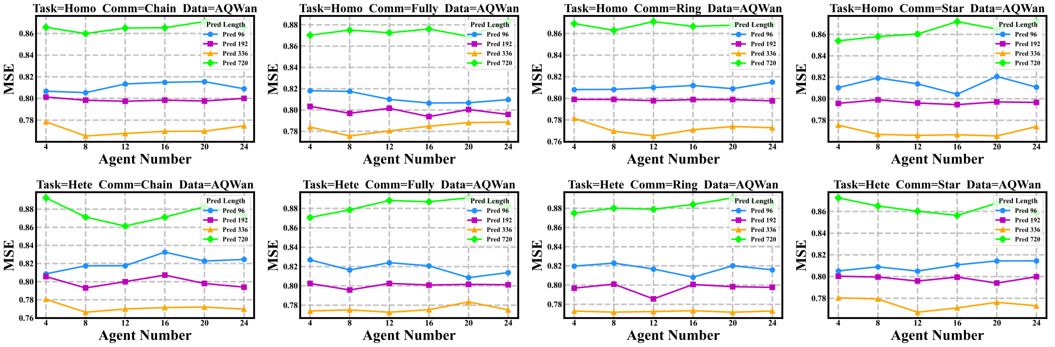

Figure 20: Performance comparison of different agent numbers on AQWan.

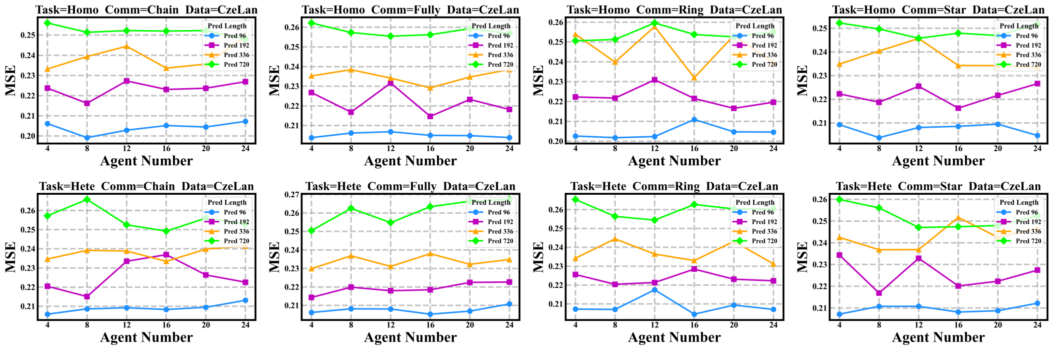

Figure 21: Performance comparison of different agent numbers on CzeLan.

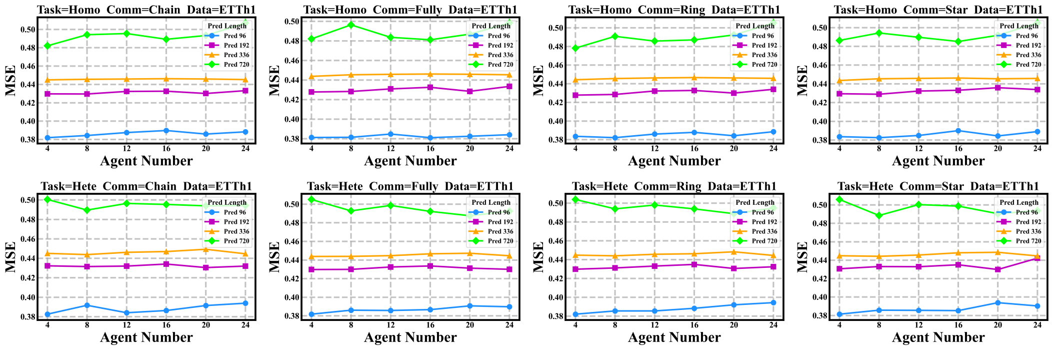

Figure 22: Performance comparison of different agent numbers on ETTh1.

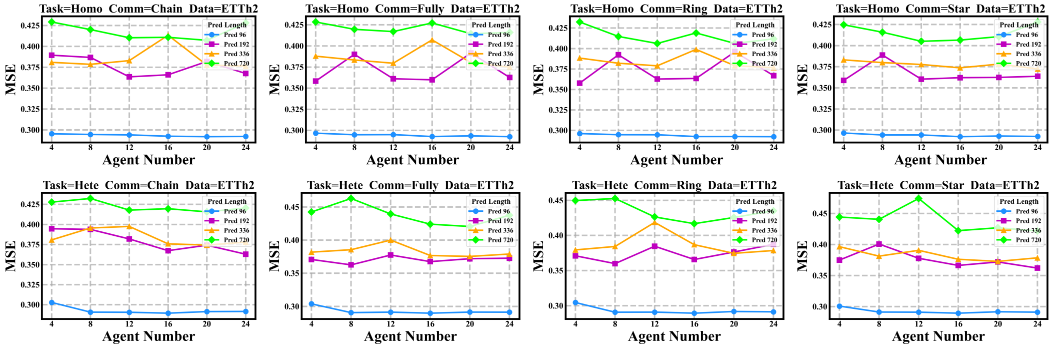

Figure 23: Performance comparison of different agent numbers on ETTh2.

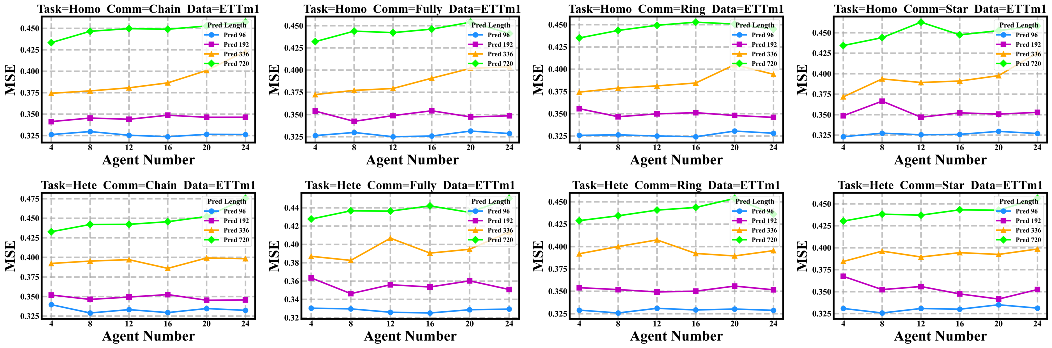

Figure 24: Performance comparison of different agent numbers on ETTm1.

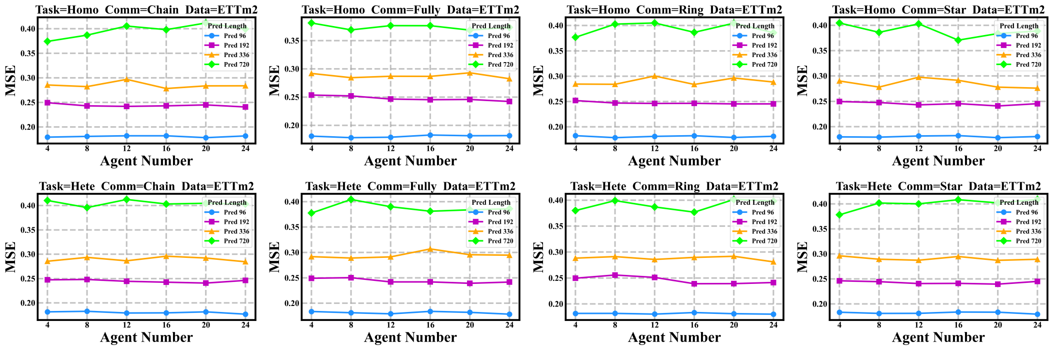
Task=Homo_Comm=Chain_Data=ETTm2
Task=Homo_Comm=Fully_Data=ETTm2
Task=Homo_Comm=Ring_Data=ETTm2
Task=Homo_Comm=Star_Data=ETTm2
Task=Hete_Comm=Chain_Data=ETTm2
Task=Hete_Comm=Fully_Data=ETTm2
Task=Hete_Comm=Ring_Data=ETTm2
Task=Hete_Comm=Star_Data=ETTm2

Figure 25: Performance comparison of different agent numbers on ETTm2.

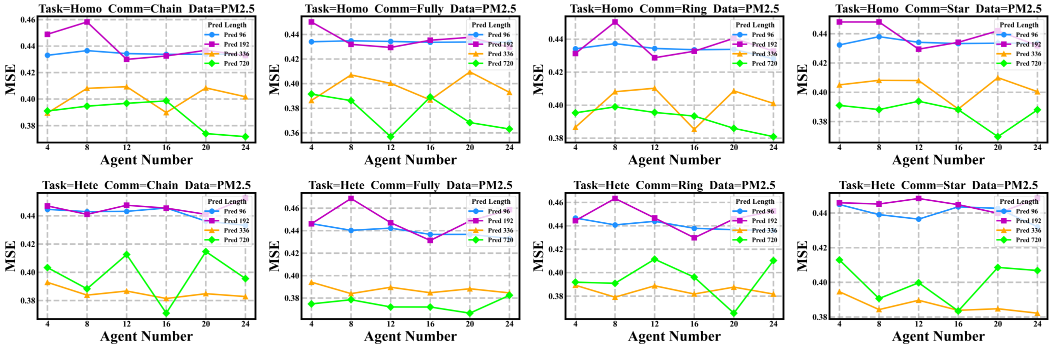
Task=Homo_Comm=Chain_Data=PM2.5
Task=Homo_Comm=Fully_Data=PM2.5
Task=Homo_Comm=Ring_Data=PM2.5
Task=Homo_Comm=Star_Data=PM2.5
Task=Hete_Comm=Chain_Data=PM2.5
Task=Hete_Comm=Fully_Data=PM2.5
Task=Hete_Comm=Ring_Data=PM2.5
Task=Hete_Comm=Star_Data=PM2.5

Figure 26: Performance comparison of different agent numbers on pm2.5.

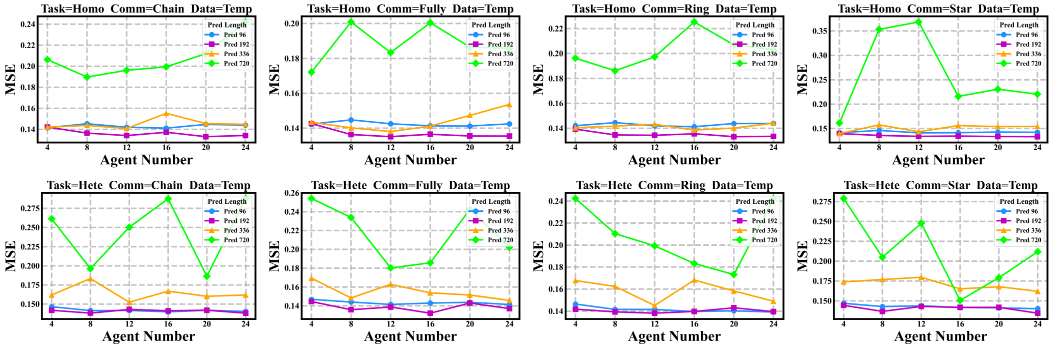
Task=Homo_Comm=Chain_Data=Temp
Task=Homo_Comm=Fully_Data=Temp
Task=Homo_Comm=Ring_Data=Temp
Task=Homo_Comm=Star_Data=Temp
Task=Hete_Comm=Chain_Data=Temp
Task=Hete_Comm=Fully_Data=Temp
Task=Hete_Comm=Ring_Data=Temp
Task=Hete_Comm=Star_Data=Temp

Figure 27: Performance comparison of different agent numbers on temp.

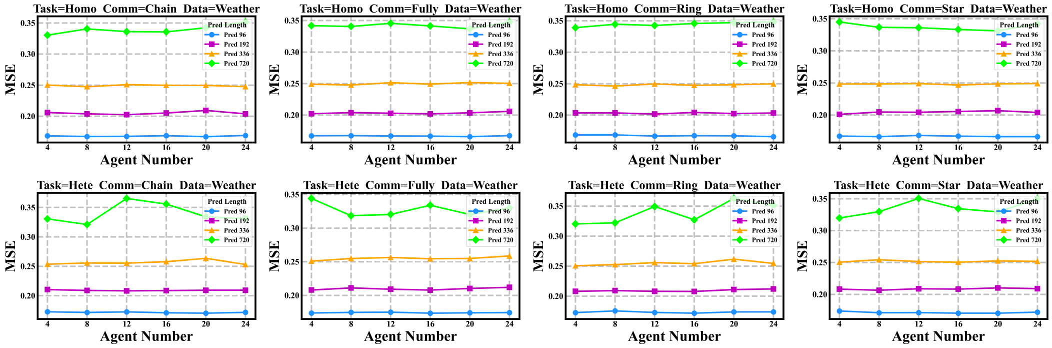
Task=Homo_Comm=Chain_Data=Weather
Task=Homo_Comm=Fully_Data=Weather
Task=Homo_Comm=Ring_Data=Weather
Task=Homo_Comm=Star_Data=Weather
Task=Hete_Comm=Chain_Data=Weather
Task=Hete_Comm=Fully_Data=Weather
Task=Hete_Comm=Ring_Data=Weather
Task=Hete_Comm=Star_Data=Weather

Figure 28: Performance comparison of different agent numbers on weather.

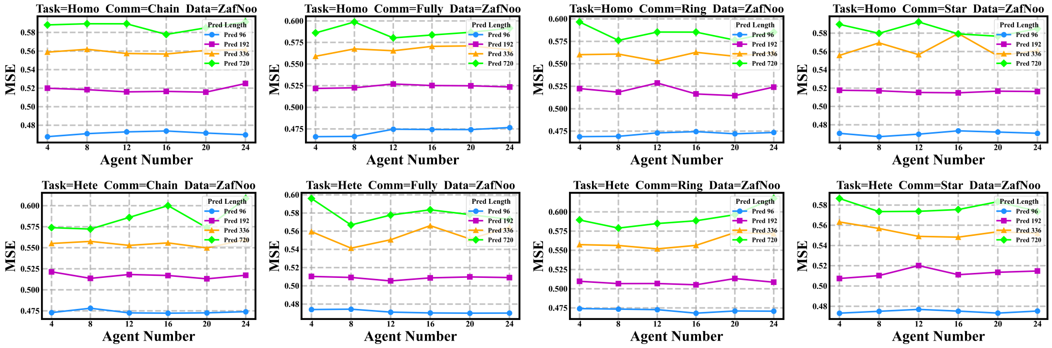

Figure 29: Performance comparison of different agent numbers on ZafNoo.

F Limitation

This work introduces MAFS, a novel multi-agent forecasting system that successfully leverages collective intelligence to enhance long-term time series forecasting. Through specialized agent design, structured communication, and adaptive decision fusion, MAFS achieves consistent performance improvements across diverse datasets and forecasting horizons, demonstrating its strong generalization and robustness. However, despite these promising results, multi-agent forecasting still faces certain limitations. Specifically, the system exhibits occasional instability during training, particularly when the number of agents increases or when dealing with highly volatile datasets. This instability may arise from conflicting agent objectives or suboptimal communication structures that hinder effective information integration.

In future work, we plan to further improve MAFS from the following perspectives:

- **Adaptive Communication Topologies**: Develop dynamic topology learning mechanisms that can automatically adjust inter-agent connections based on data characteristics and task complexity, reducing reliance on manually defined structures.
- **Diverse Task Decomposition Strategies**: Explore more fine-grained and semantically meaningful task divisions, enabling agents to specialize in richer forecasting aspects such as uncertainty estimation, anomaly detection, or rare event prediction.

We believe these directions will further enhance the stability, adaptability, and forecasting capability of multi-agent systems in real-world scenarios.

G Full Long-term Time Series Forecasting Results

Table 4: Comparison of long-term time series forecasting methods on 11 datasets using MSE and MAE (lower is better). Best results are marked in **red**; second-best results are underlined in blue.

Methods		MAFS (Ours)		iTransformer [2024]		TimeMixer [2024]		PatchTST [2024]		Crossformer [2023]		TiDE [2024]		TimesNet [2023]		DLinear [2023]		Autoformer [2021]		Informer [2021]	
Datasets	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.381	**0.399**	0.392	0.41	**0.375**	0.4	0.414	0.419	0.423	0.448	0.479	0.464	0.384	0.402	0.386	0.4	0.449	0.459	0.865	0.713
	192	**0.428**	0.424	0.442	0.442	0.429	**0.421**	0.46	0.445	0.471	0.474	0.525	0.492	0.436	0.429	0.437	0.432	0.5	0.482	1.008	0.792
	336	**0.444**	**0.442**	0.483	0.473	0.484	0.458	0.501	0.466	0.57	0.546	0.565	0.515	0.491	0.469	0.481	0.459	0.521	0.496	1.107	0.809
	720	**0.478**	0.484	0.552	0.537	0.498	**0.482**	0.5	0.488	0.653	0.621	0.594	0.558	0.521	0.5	0.519	0.516	0.514	0.512	1.181	0.865
	Avg	**0.433**	**0.437**	0.467	0.466	0.447	0.44	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.45	0.456	0.452	0.496	0.487	1.04	0.795
ETTh2	96	**0.289**	**0.341**	0.304	0.353	0.289	0.341	0.302	0.348	0.745	0.584	0.4	0.44	0.34	0.374	0.333	0.387	0.346	0.388	3.755	1.525
	192	**0.358**	**0.391**	0.399	0.417	0.372	0.392	0.388	0.4	0.877	0.656	0.528	0.509	0.402	0.414	0.477	0.476	0.456	0.452	5.602	1.931
	336	**0.372**	**0.406**	0.41	0.429	0.386	0.414	0.426	0.433	1.043	0.731	0.643	0.571	0.452	0.452	0.594	0.541	0.482	0.486	4.721	1.835
	720	**0.405**	0.438	0.433	0.463	0.412	**0.434**	0.431	0.446	1.104	0.763	0.874	0.679	0.462	0.468	0.831	0.657	0.515	0.511	3.647	1.625
	Avg	**0.356**	**0.394**	0.386	0.415	0.364	0.395	0.387	0.407	0.942	0.684	0.611	0.55	0.414	0.427	0.559	0.515	0.45	0.459	4.431	1.729
ETTm1	96	0.323	0.361	0.336	0.372	**0.32**	**0.357**	0.329	0.367	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372	0.505	0.475	0.672	0.571
	192	**0.341**	**0.373**	0.366	0.39	0.361	0.381	0.367	0.385	0.45	0.451	0.398	0.404	0.374	0.387	0.38	0.389	0.553	0.496	0.795	0.669
	336	**0.372**	**0.393**	0.388	0.41	0.39	0.404	0.399	0.41	0.532	0.515	0.428	0.425	0.41	0.411	0.413	0.413	0.621	0.537	1.212	0.871
	720	**0.428**	**0.426**	0.442	0.442	0.454	0.441	0.454	0.439	0.666	0.589	0.487	0.461	0.478	0.45	0.474	0.453	0.671	0.561	1.166	0.823
	Avg	**0.366**	**0.388**	0.383	0.403	0.381	0.395	0.387	0.4	0.513	0.496	0.419	0.419	0.4	0.406	0.403	0.407	0.588	0.517	0.961	0.734
ETTm2	96	0.177	0.26	0.184	0.266	**0.175**	**0.258**	0.175	0.259	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292	0.255	0.339	0.365	0.453
	192	0.239	0.302	0.256	0.317	**0.237**	**0.299**	0.241	0.302	0.414	0.492	0.29	0.364	0.249	0.309	0.284	0.362	0.281	0.34	0.533	0.563
	336	**0.276**	**0.329**	0.313	0.355	0.298	0.34	0.305	0.343	0.597	0.542	0.377	0.422	0.321	0.351	0.369	0.427	0.339	0.372	1.363	0.887
	720	**0.369**	**0.395**	0.407	0.417	0.391	0.396	0.402	0.4	1.73	1.042	0.558	0.524	0.408	0.403	0.554	0.522	0.433	0.432	3.379	1.338
	Avg	**0.265**	**0.321**	0.29	0.339	0.275	0.323	0.281	0.326	0.757	0.61	0.358	0.404	0.291	0.333	0.35	0.401	0.327	0.371	1.41	0.81
Weather	96	0.166	**0.206**	0.175	0.215	0.163	0.209	0.177	0.218	**0.158**	0.23	0.202	0.261	0.172	0.22	0.196	0.255	0.266	0.336	0.3	0.384
	192	**0.201**	**0.244**	0.214	0.254	0.208	0.25	0.225	0.259	0.206	0.277	0.242	0.298	0.219	0.261	0.237	0.296	0.307	0.367	0.598	0.544
	336	**0.246**	**0.282**	0.252	0.288	0.251	0.287	0.278	0.297	0.272	0.335	0.287	0.335	0.28	0.306	0.283	0.335	0.359	0.395	0.578	0.523
	720	**0.318**	**0.338**	0.331	0.348	0.339	0.341	0.354	0.348	0.398	0.418	0.351	0.386	0.365	0.359	0.345	0.381	0.419	0.428	1.059	0.741
	Avg	**0.233**	**0.267**	0.243	0.276	0.24	0.271	0.259	0.281	0.259	0.315	0.271	0.32	0.259	0.287	0.265	0.317	0.338	0.382	0.634	0.548
AQShunyi	96	0.711	0.499	0.742	0.506	0.731	0.533	**0.648**	**0.481**	0.652	0.484	0.708	0.52	0.658	0.488	0.651	0.492	0.736	0.529	0.754	0.542
	192	0.697	0.502	0.71	0.507	0.711	**0.467**	0.69	0.501	**0.674**	0.499	0.774	0.569	0.707	0.511	0.691	0.512	0.735	0.535	0.759	0.536
	336	**0.668**	**0.506**	0.687	0.51	0.684	0.564	0.711	0.515	0.704	0.515	0.827	0.56	0.785	0.537	0.716	0.529	0.83	0.566	0.837	0.56
	720	**0.728**	0.533	0.752	0.539	0.749	0.554	0.77	0.538	0.747	**0.518**	0.803	0.566	0.755	0.527	0.765	0.556	0.754	0.532	0.777	0.543
	Avg	0.701	0.509	0.723	0.515	0.719	0.529	0.705	0.509	**0.694**	**0.504**	0.778	0.554	0.726	0.516	0.706	0.522	0.764	0.541	0.782	0.545
AQWan	96	0.804	0.49	0.814	0.491	0.829	**0.456**	**0.745**	0.47	0.75	0.465	0.833	0.524	0.791	0.488	0.756	0.481	0.858	0.518	0.901	0.522
	192	0.786	0.496	0.801	0.497	0.81	0.501	0.792	0.491	**0.762**	**0.479**	0.82	0.516	0.779	0.49	0.8	0.502	0.803	0.513	0.833	0.521
	336	**0.765**	**0.496**	0.786	0.502	0.791	0.538	0.819	0.503	0.802	0.504	0.858	0.552	0.814	0.505	0.823	0.516	0.826	0.523	0.847	0.525
	720	0.854	0.529	0.868	0.536	0.883	**0.499**	0.89	0.533	**0.83**	0.511	0.913	0.551	0.869	0.519	0.891	0.548	0.872	0.547	0.883	0.532
	Avg	0.802	0.503	0.817	0.507	0.828	0.499	0.812	0.499	**0.786**	**0.49**	0.856	0.536	0.813	0.5	0.818	0.512	0.84	0.525	0.866	0.525
CzeLan	96	0.199	0.248	0.21	0.255	0.202	0.263	0.183	0.251	0.581	0.443	0.186	0.256	**0.176**	**0.237**	0.211	0.289	0.238	0.294	0.25	0.305
	192	0.214	**0.258**	0.231	0.275	0.22	0.288	**0.208**	0.271	0.705	0.503	0.226	0.29	0.215	0.279	0.252	0.323	0.29	0.341	0.295	0.337
	336	0.229	0.281	0.243	0.293	0.237	**0.266**	0.243	0.302	0.971	0.596	0.238	0.304	**0.224**	0.288	0.317	0.366	0.322	0.357	0.335	0.361
	720	0.246	0.298	**0.245**	**0.297**	0.254	0.302	0.273	0.335	1.566	0.762	0.295	0.363	0.282	0.337	0.358	0.392	0.379	0.427	0.384	0.416
	Avg	**0.222**	**0.271**	0.232	0.28	0.228	0.28	0.227	0.29	0.956	0.576	0.237	0.303	0.224	0.285	0.284	0.342	0.307	0.355	0.316	0.355
ZafNoo	96	0.466	0.411	0.476	0.419	0.481	**0.404**	0.444	0.426	**0.432**	0.419	0.508	0.45	0.479	0.424	0.434	0.411	0.524	0.468	0.541	0.473
	192	0.505	**0.439**	0.529	0.457	0.527	0.454	0.498	0.456	**0.479**	0.449	0.536	0.491	0.491	0.446	0.484	0.444	0.687	0.558	0.708	0.575
	336	0.541	0.465	0.568	0.488	0.56	**0.444**	0.53	0.48	0.521	0.469	0.592	0.519	0.551	0.479	**0.518**	0.464	0.835	0.669	0.851	0.661
	720	0.567	0.494	0.591	0.509	0.585	**0.46**	0.574	0.499	**0.543**	0.483	0.642	0.533	0.627	0.511	0.548	0.486	0.854	0.702	0.876	0.699
	Avg	0.52	0.451	0.541	0.468	0.538	**0.44**	0.511	0.465	**0.494**	0.455	0.569	0.498	0.537	0.465	0.496	0.451	0.725	0.599	0.744	0.602
PM2.5	96	**0.428**	**0.428**	0.438	0.432	0.446	0.48	0.46	0.468	0.451	0.46	0.475	0.484	0.481	0.481	0.453	0.466	0.525	0.512	0.581	0.594
	192	**0.429**	0.425	0.436	0.429	0.45	**0.408**	0.462	0.449	0.463	0.446	0.493	0.476	0.455	0.446	0.447	0.431	0.519	0.502	0.55	0.537
	336	**0.379**	**0.404**	0.412	0.417	0.395	0.441	0.466	0.505	0.447	0.469	0.455	0.473	0.469	0.508	0.451	0.485	0.53	0.553	0.524	0.546
	720	**0.357**	**0.399**	0.4	0.408	0.37	0.414	0.452	0.493	0.464	0.514	0.499	0.554	0.49	0.535	0.458	0.524	0.485	0.53	0.499	0.566
	Avg	**0.398**	**0.414**	0.421	0.421	0.415	0.436	0.46	0.479	0.456	0.472	0.481	0.497	0.473	0.492	0.453	0.477	0.515	0.524	0.539	0.561
Temp	96	0.139	**0.282**	**0.138**	0.283	0.144	0.306	0.144	0.3	0.186	0.377	0.155	0.323	0.162	0.329	0.152	0.317	0.185	0.373	0.197	0.405
	192	**0.132**	**0.279**	0.145	0.296	0.137	0.317	0.145	0.309	0.203	0.435	0.157	0.323	0.173	0.374	0.155	0.322	0.213	0.45	0.199	0.429
	336	**0.138**	**0.289**	0.203	0.352	0.145	0.297	0.147	0.313	0.209	0.435	0.169	0.357	0.185	0.387	0.162	0.342	0.231	0.492	0.243	0.506
	720	**0.151**	0.3	0.204	0.352	0.157	**0.298**	0.153	0.303	0.224	0.445	0.173	0.35	0.312	0.623	0.167	0.329	0.347	0.683	0.311	0.627
	Avg	**0.14**	**0.288**	0.173	0.321	0.146	0.304	0.147	0.306	0.206	0.423	0.164	0.338	0.208	0.428	0.159	0.327	0.244	0.5	0.238	0.492

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions and scope of the paper. In particular, the introduction 1 outlines the motivation for applying multi-agent systems to time series forecasting, explicitly presents the key challenges, and summarizes the core solutions proposed by MAFS, including sub-task decomposition, inter-agent communication, and voting aggregation. These claims are consistent with the overall scope and objectives of the work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present the limitations of this work and potential future directions in Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not include theoretical assumptions and results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed descriptions of the model architecture, datasets, and implementation. Additionally, the source code is released to facilitate reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have uploaded the source code to an anonymous repository, and all datasets used are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and testing details, including data splits, hyperparameters, and optimizer settings, are provided in Section 5.1 and Section C to ensure clarity and reproducibility of the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For all experimental results in this paper we provided the mean of the numerous experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided basic information about our deployment platform and computing power in Section C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We make sure our research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The forecasting agents in our MAFS framework are based on the open-sourced model iTransformer. We use the official implementation available at https://github.com/thuml/iTransformer, which is released under the MIT License. We properly cite the original paper that proposed iTransformer in our manuscript. All usage of the code strictly follows the terms of the MIT License. Additionally, all datasets used in our experiments are publicly available and used in accordance with their respective licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.