Cross-MoE: An Efficient Temporal Prediction Framework Integrating Textual Modality

Ruizheng Huang
University of Electronic Science and Technology of China
huangrz@std.uestc.edu.cn

Zhicheng Zhang
University of Electronic Science and Technology of China
zhangzc@stu.uestc.edu.cn

Yong Wang
University of Electronic Science and Technology of China
cla@uestc.edu.cn

Abstract

It has been demonstrated that incorporating external information as textual modality can effectively improve time series forecasting accuracy. However, current multi-modal models ignore the dynamic and different relations between time series patterns and textual features, which leads to poor performance in temporal-textual feature fusion. In this paper, we propose a lightweight and model-agnostic temporal-textual fusion framework named Cross-MoE. It replaces Cross Attention with Cross-Ranker to reduce computational complexity, and enhances modality-aware correlation memorization with Mixture-of-Experts (MoE) networks to tolerate the distributional shifts in time series. The experimental results demonstrate a 8.78% average reduction in Mean Squared Error (MSE) compared to the SOTA multi-modal time series framework. Notably, our method requires only 75% of computational overhead and 12.5% of activated parameters compared with Cross Attention mechanism. Our codes are available at https://github.com/Kilosigh/Cross-MoE.git

1 Introduction

Time series forecasting (TSF) plays a crucial role across various fields, such as financial market analysis (Sezer et al., 2020), energy demand management (Deb et al., 2017) and healthcare monitoring(Kaushik et al., 2020). By analyzing historical data and identifying underlying patterns, TSF can help reveal potential trends, cyclical changes and anomalies, thereby effectively supporting strategic planning and resource allocation.

However, in many real-world scenarios, such as stock markets, agriculture and energy consumption, external factors like policy changes (Lencucha et al., 2020; Hirschman and Berman, 2014), expert opinions (Leal et al., 2007; Kamali et al., 2017) can significantly influence the future behaviors of time series. Correspondingly, their distributions often exhibit temporal variations, which is a phenomenon known as Temporal Distribution Shift (TDS) (Fan et al., 2023). Such non-stationarity is often introduced by external factors. Hence, it is not enough to only analyze numerical time series data alone because its patterns inherently lack explicit indicators of the external influences.

Figure 1: An example of the temporal distribution shift correlated with textual information.

Figure 1 shows the TDS phenomenon in Energy prices. The three temporal intervals of significant pattern shifts are annotated with their corresponding textual descriptions that are provided by Reuters U.S reports and ScienceDirect. Consequently, there is a growing recognition (Liu et al., 2024b; Zhang et al., 2024b; Kim et al., 2024) that integrating textual information with numerical time series data can enhance the forecasting ability.

Unfortunately, existing temporal-textual fusion approaches are oversimplified. GPT4MTS(Tao et al., 2024) concatenate textual features generated by large language models (LLMs) with time series features to improve TSF accuracy. Time-MMD (Liu et al., 2024b) uses weighted summation of modal-specific forecasting results to achieve modality fusion. These strategies fail to systematically evaluate the inter-modal correlations and contribution weights between textual and temporal information. Moreover, the text embeddings employed in these methods are typically generated by

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 29927–29938
November 4-9, 2025 ©2025 Association for Computational Linguistics

Large Language Models (LLMs), which primarily capture semantic information rather than temporal context critical for TSF. To address such modality misalignment, recent work (Xu et al., 2024) proposes leveraging Cross Attention mechanisms to filter irrelevant textual content. Their model employs time series patch-wise features as queries, with textual features serving as keys and values, thereby simultaneously filtering out irrelevant information in texts and achieving modality fusion. It introduces substantial computation overhead that limits its practical deployment.

In this paper, we propose a model-agnostic temporal-textual fusion framework, which consists of MoE networks and a Cross-Ranker. The MoE is used to learn the correspondence between texts tokens and time series patterns, while the Cross-Ranker is responsible for filtering out irrelevant information from the text and synchronizing the text with forecast lengths. Our main contributions can be summarized as follows:

1. We propose Cross-MoE, a model-agnostic temporal-textual fusion framework that decouples temporal models into an encoder and a decoder/projection head. The framework integrates textual features with encoder-derived temporal features while preserving dimensional consistency, then feeds the fused representations into the decoder/projection head to generate final predictions without requiring structural modifications to the original temporal model architecture.

2. Our framework employs MoE networks to dynamically select and project features of temporal interval with distinct patterns alongside their corresponding textual tokens. A Cross-Ranker subsequently filters these projected representations based on similarity, enabling targeted fusion of aligned features while filtering out irrelevant information from the textual modality.

3. We test various modality fusion approaches on the nine multi-modal time series Time-MMD (Liu et al., 2024b) datasets. Our results show that Cross-MoE lead to an average reduction of 8.78% in MSE compared to the SOTA multi-modal time series framework. Additionally, it achieves better performance than Cross Attention, which consumes only 75% of the computational cost and 12.5% of the activated parameters.

2 Related Works

2.1 Time series models

Time series forecasting models have undergone significant development over time. Early approaches like ARIMA (AutoRegressive Integrated Moving Average)(Box and Pierce, 1970) were designed to model linear time series by capturing temporal dependencies through autoregressive and moving average components. As time series data became more complex, Recurrent Neural Networks (RNNs) (Cao et al., 2018; Yoon et al., 2018) were introduced to better capture sequential dependencies and non-linear patterns. Following this, Convolutional Neural Networks (CNNs) (Bai et al., 2018; Luo and Wang, 2024) offer an effective approach for their ability to capture local features and pattern recognition.

More recently, Transformer-based models have gained widespread use due to their ability to capture long-range dependencies and their parallelized training process. Autoformer(Wu et al., 2021) uses trend and seasonal decomposition along with a sub-quadratic self-attention mechanism. FEDFormer (Zhou et al., 2022) integrates a frequency-enhanced structure, while Pyraformer (Liu et al.) adopts pyramidal self-attention to achieve linear complexity and capture both short and long temporal dependencies.

Since point-wise representations are limited in capturing local semantic patterns within temporal variations (Zeng et al., 2023), PatchTST (Nie et al., 2022) replaces the time points with segmented subseries called patches, and feeds the tokens of which to the vanilla self-attention mechanism. Beyond capturing the patch-level temporal dependencies within one single series, recent approaches have endeavored to capture interdependencies among patches from different variables over time. Crossformer (Zhang and Yan, 2023) introduces a Two-Stage Attention layer to efficiently capture the cross-time and cross-variate dependencies of each patch. Further expanding the receptive field, iTransformer (Liu et al., 2024c) utilizes the global representation of the whole series and applies attention to these series-wise representations to capture multivariate correlations. Timesnet(Wu et al., 2022) captures both intraperiod and interperiod relationships simultaneously in the time series

data converted into 2D.

Building on these advancements, our work proposes a multimodal framework for Transformer-based models that integrates textual modality information with the TS-Encoder, generating enhanced prediction results.

2.2 Text-assisted Time series prediction

2.2.1 TS2Text methods

Several studies have explored converting time series data or its statistical summaries into text to input into LLMs for prediction tasks. Time-CMA(Liu et al., 2024a) converts the original data along with its first-order differences into text and inputs it into a large model, using a cross-attention mechanism to perform modality fusion. Time-LLM(Jin et al., 2024) incorporates statistical information such as the minimum, maximum, and mean values of the data, along with relevant background information about the dataset, to enhance the model's understanding. UniMTS(Zhang et al., 2024a), on the other hand, employs a contrastive learning approach to align action time series features with their corresponding textual descriptions, selecting the most relevant text based on the similarity between the features. These models typically do not incorporate information beyond the original time series data, except for a small amount of auxiliary text included in the prompt by the authors. Nonetheless, they still demonstrate the potential of leveraging textual information to improve time series forecasting by enhancing the model's contextual understanding and predictive power.

2.2.2 TS+Text methods

Studies have been explored to incorporate exogenous textual information, such as news articles, policies, and expert opinions, to assist time series models in forecasting. Time-MMD(Liu et al., 2024b) created a dataset for such tasks, where the final prediction is obtained by combining the results of a large model's prediction based solely on textual information with the prediction from a time series model through a weighted sum. GPT4MTS(Jia et al., 2024) adds the embeddings from both modalities together and inputs them into an LLM to generate the final prediction. TGTSF(Xu et al., 2024) calculates the similarity between future news information and channel descriptions to obtain the textual modality embeddings, which are then combined with the output from the TS-Encoder through a cross-attention mechanism to produce the final prediction. In contrast, our work uses MoE and similarity calculations to classify, project, and filter the textual information, enabling efficient and effective modality fusion in time series forecasting.

3 Preliminaries

3.1 Problem Definition

Given a dataset contains time series and corresponding textual sentences, $\mathcal{D} = \{X^i_{t-L:t}, X^i_{t:t+H}, S^i_t\}_{i=1}^{|\mathcal{D}|}$, where the sequence $X^i_{t-L:t} \in \mathbb{R}^{L\times C}$ denotes the input time series, L is the length of look-back window and C represents the number of channels (variables). S^i_t is the text corresponding to the $X^i_{t-L:t} \in \mathbb{R}^{L\times C}$. Our objective is to find a function $f_\theta(X^i_{t-L:t}, S^i_t)$ parameterized by θ to minimize the mean square errors between the ground truths $X^i_{t:t+H}$ and forecasting results $\hat{X}^i_{t:t+H}$, i.e.

$$\underset{\theta}{argmin}\{\frac{1}{|\mathcal{D}|}\sum_{i=1}^{|\mathcal{D}|}\|\hat{X}^i_{t:t+H} - X^i_{t:t+H}\|_F^2\} \quad (1)$$

4 Method

Our method utilizes the temporal features generated by the time series (TS) model's encoder and the textual features generated by the LLM for modality fusion. It mainly consists of two modules: Cross-Ranker and MoE. The Cross-Ranker module filters out irrelevant information in the text modality and aligns the token numbers of the text modality with the time series data. The MoE network is responsible for learning the correspondence between the patterns in the time series data and the text tokens, projecting the token features into the time series feature space. The architecture is shown in Figure 2, where the modality fusion process is demonstrated based on point-wise approach. The subsequent method explanations will also be based on the Patch-based approach.

4.1 Time series Encoding

Our framework treats the time series model as a white-box model, decomposing it into two components: the Encoder and the Decoder (which could also be a Projection Head). The Encoder is responsible for extracting features from the given time series data, while the Head uses these features to make predictions. We leverage the outputs of Encoder for fusion of text modality.

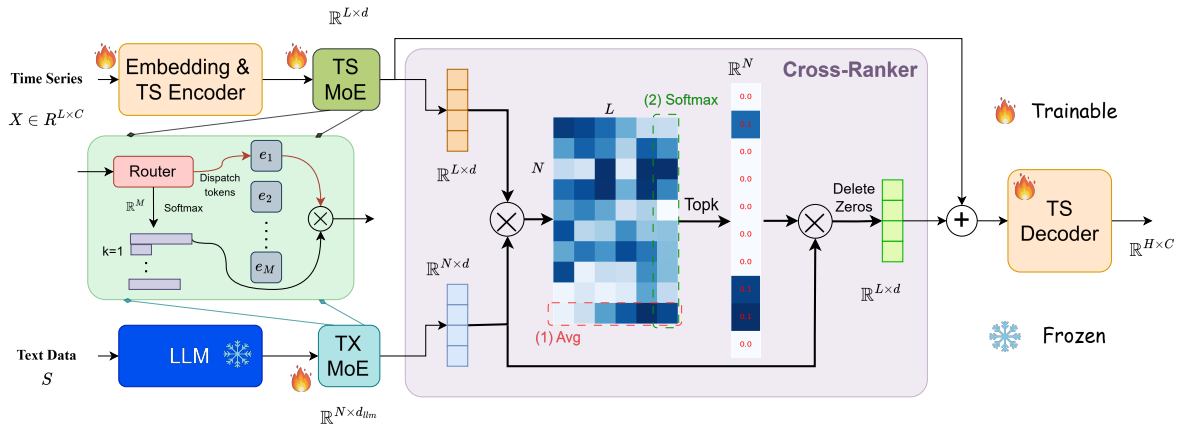

Figure 2: The architecture of Cross-MoE. The Cross-Ranker is used to select important textural features. The TS MoE and TX MoE are used to memorize the correlation between time series patterns and textual features.

Let $X \in \mathbb{R}^{L \times C}$ denote the input time series, and the L represents the length of the look-back window and C denotes the number of channels (variables). After encoding through a TS model, let the output representation be denoted as:

$$X^{en} = Encode(X),\ X^{en} \in \mathbb{R}^{L \times d} \tag{2}$$

where d is the dimensionality of time series embedding.

Specifically, the encoded representation X^{en} generated by patch-based temporal models should maintain a shape of $\mathbb{R}^{C \cdot PN_{lbw} \times d}$, where PN_{lbw} denotes the number of patches corresponding to the look-back window L.

4.2 Mixture of Experts network

As shown in Figure 2, MoE networks introduce multiple expert models, which allow the system to dynamically select the most relevant experts based on the input. Such selective computation helps expand the model's capacity to capture and store information, while only a small subset of experts is activated for each forward pass, thereby reducing computational cost.

A single MoE network consists of a set of experts $E = \{e_i\}_{i=1}^{M}, e_i : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ and a router network $G(\cdot) : \mathbb{R}^{d_{in}} \to \mathbb{R}^{M}$, where d_{in} and d_{out} denotes the embedding dimension of input and output tokens, respectively. Each expert is a simple MLP layer. The router network is responsible for distributing the embedding of each time step token to one or more experts as input. The calculation process of a MoE network could be defined as :

$$\text{MoE}(x) = \sum_{i=1}^{M} G_i(x) \cdot e_i(x) \tag{3}$$

$$G_i(x) = \begin{cases} s_i, & \text{if } s_i \in \text{TopK}_{j=1}^{M}(s_j, k), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$s_i = \text{Softmax}(W_g \cdot x)_i \tag{5}$$

where $W_g \in \mathbb{R}^{M \times d}$ denotes trainable parameters, M is the number of experts and k represents the hyperparameter in the TopK selection function, indicating the number of experts to be selected.

4.3 Cross-Ranker

The Cross-Ranker aims to simplify the computation in Cross Attention. Traditional Cross Attention computes the correlation score between the query and key by projecting them once and then performing a dot product between the output vectors. We adopt a similar approach for computing the scores. However, the key difference is that we retain only this step and select the top-k text tokens with the highest correlation scores as the output. This not only significantly reduces the computational and memory overhead, but achieves performance comparable to Cross Attention when combined with the MoE network.

When the text data is input, it is first processed by a LLM to obtain the Text Embedding $S^{en} \in \mathbb{R}^{N \times d_{llm}}$, where N represents the sentence length and d_{llm} is the feature dimension of the text tokens. Subsequently, both temporal and textual features are processed through MoE networks, which produces $X^{en'} \in \mathbb{R}^{L \times d}$ and $S^{en'} \in \mathbb{R}^{N \times d}$. Such operation achieves dual objectives:

(1) Dimensional alignment between heterogeneous modalities.

(2) Joint projection into a unified latent space that preserves cross-modal semantic relationships.

$$S^{en'} = MoE(S^{en}), \quad X^{en'} = MoE(X^{en}) \quad (6)$$

Such an operation ensures that the subsequent similarity calculation accurately reflects the informational relevance between the time series and text data. A similarity matrix $A \in \mathbb{R}^{N \times L}$ is calculated, which could be described as:

$$A = S^{en'} \times (X^{en'})^T \quad (7)$$

The next step is to rank the tokens based on similarity, thereby filtering out the useless information from the text to obtain $S^{fu} \in \mathbb{R}^{L \times d}$:

$$S^{fu} = \{w_i \cdot S_i^{en'}, w_i \in TopK(w, L)\} \quad (8)$$

$$w_i = \frac{e^{u_i}}{\sum_j e^{u_j}}, \quad u_i = \sum_j A_{ij} \quad (9)$$

where A_{ij} denotes the similarity between the i-th text token and j-th time series patch and A is the result of Eq. 7.

4.4 Forecasting Generation

The decoder of the decoupled time series model is used as the forecasting generation component, which takes the summation of the filtered text features S^{fu} and the encoder output $X^{en'}$ as the input. The forecasting results $X^{out} \in \mathbb{R}^{H \times C}$ are generated by the decoder.

$$X^{out} = Decoder(S^{fu} + X^{en'}) \quad (10)$$

where H denotes the prediction length.

4.5 Total loss

The final loss of the model consists of two components: the mean squared error (MSE) between the predicted results and the ground truth labels, and an auxiliary loss introduced by the MoE network. The latter loss serves to regularize the Router, which encourages more balanced token allocation decisions and ensures a more even distribution of load across the experts. The detailed computation process is as follows:

$$\mathcal{L}^{total} = \mathcal{L}^{mse} + \lambda \cdot \mathcal{L}^{aux} \quad (11)$$

$$\mathcal{L}^{mse} = \frac{1}{C \cdot H} \sum_{i=1}^{C} \sum_{j=1}^{H} (X_{ij}^{out} - X_{ij}^{lable})^2 \quad (12)$$

$$\mathcal{L}^{aux} = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N} G_i(S_j^{en}) \quad (13)$$

The Eq.12 calculates the MSE, while Eq.13 represents the auxiliary loss. In Eq.13, the computation of $G_i(\cdot)$ is given by Eq.4, which represents the weight of the i-th expert, and S_j^{en} denotes the feature of the j-th text token. In Eq.11, λ is the hyperparameter weight coefficient that balances the importance of the two loss components.

5 Experiments

5.1 Experimental Setup

In this section, we introduce the experimental setup, including the dataset, hardware platform, and hyperparameter configurations.

Dataset: Our experiments use the Time-MMD dataset, which covers time series data and corresponding text from nine domains: Agriculture, Climate, Economy, Energy, Entertainment, Environment, Public Health, Security, Social Good and Traffic. The dataset spans three distinct frequencies: daily, weekly, and monthly. The numerical data is sourced from reliable government agencies, and the data across different domains exhibits various patterns, such as periodicity and trends. The text data comes from government reports and web searches. The entire dataset is split into training, validation, and test sets in a 7:1:2 ratio.

Hardware Platform: All experiments are conducted on a server with the following specifications: CPU: Intel(R) Xeon(R) Gold 6348, GPU: NVIDIA A8000 80GB, System: Ubuntu 22.04.3 LTS.

Hyperparameter Settings: We use a batch size of 64, the Adam optimizer, a learning rate of 5e-4. We employ a cosine schedule to dynamically adjust the learning rate, where the learning rate at the current epoch is calculated as $1 + \cos(\pi * \frac{\text{current epochs}}{\text{total epochs}})$. The early stopping tolerance is set to 5 epochs, and the dropout rate is set to 0.2. For all tasks, the number of experts $M \in \{4, 8, 16, 32\}$.

5.2 Main Results

We use the Time-MMD framework as the baseline and employ PatchTST(Nie et al., 2022), iTransformer(Liu et al., 2024c), TimeXer(Wang et al.,

Table 1: Comparison of performance between different models in our framework and the Time-MMD framework. Lower MSE/MAE values indicate better predictive ability. The best result across frameworks is captioned **bold**.

Model		TimeXer		TimesNet		PatchTST		iTransformer		Reformer		TimeLLM	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Dataset	Arch												
Algriculture	Uni-modal	0.13310	0.27107	0.17338	0.32076	0.11563	0.24940	0.09791	0.21336	0.44290	0.50787	0.12370	0.26270
	Time-MMD	0.12870	0.26662	0.13944	0.28002	0.12133	0.25616	0.11427	0.23747	0.48802	0.53759	0.13118	0.26809
	Cross-MoE	0.12737	0.25810	0.13576	0.27913	0.11360	0.23478	0.10357	0.22532	0.34049	0.42494	0.12527	0.24746
	Promotion	1.03%	3.20%	2.64%	0.32%	6.37%	8.34%	9.36%	5.11%	30.23%	20.96%	4.51%	7.70%
Climate	Uni-modal	1.08026	0.84405	1.13415	0.85276	1.23194	0.89498	1.18115	0.86385	0.87022	0.74561	1.28141	0.90856
	Time-MMD	1.11742	0.85800	1.12172	0.84773	1.12132	0.85788	1.11115	0.84622	1.02540	0.81231	1.12802	0.85383
	Cross-MoE	1.04499	0.82197	1.04377	0.81655	1.09610	0.84267	1.09627	0.84284	1.01226	0.80639	1.08025	0.83615
	Promotion	6.48%	4.20%	6.95%	3.68%	2.25%	1.77%	1.34%	0.40%	1.28%	0.73%	4.24%	2.07%
Economy	Uni-modal	0.01594	0.10125	0.02522	0.13068	0.01641	0.10315	0.01424	0.09419	0.81540	0.81655	0.02371	0.12503
	Time-MMD	0.01734	0.10572	0.03011	0.14087	0.01794	0.10605	0.01448	0.09642	0.68601	0.75139	0.02649	0.13010
	Cross-MoE	0.01412	0.09548	0.02509	0.12002	0.01076	0.08312	0.01174	0.08779	0.45721	0.59648	0.01787	0.10731
	Promotion	18.54%	9.69%	16.66%	14.80%	40.00%	21.62%	18.93%	8.96%	33.35%	20.62%	32.55%	17.52%
Energy	Uni-modal	0.25273	0.36735	0.29702	0.40554	0.25881	0.36541	0.25106	0.36742	0.45252	0.51908	0.28469	0.39427
	Time-MMD	0.26773	0.39060	0.27607	0.39346	0.26020	0.37174	0.25345	0.36637	0.46969	0.52344	0.27533	0.39275
	Cross-MoE	0.24465	0.36307	0.20400	0.32982	0.25052	0.36160	0.25911	0.37289	0.25000	0.49612	0.26945	0.38370
	Promotion	8.62%	7.05%	26.10%	16.17%	3.72%	2.73%	-2.23%	-1.78%	46.77%	5.22%	2.13%	2.30%
Environment	Uni-modal	0.43829	0.49083	0.43619	0.48801	0.56649	0.54104	0.44467	0.49504	0.49224	0.53914	0.58454	0.54959
	Time-MMD	0.43405	0.48539	0.47720	0.51254	0.51570	0.51247	0.43136	0.48586	0.47019	0.53120	0.54487	0.53171
	Cross-MoE	0.42302	0.48209	0.48635	0.50834	0.50997	0.50959	0.42617	0.48266	0.43362	0.50086	0.46383	0.50127
	Promotion	2.54%	0.68%	-1.92%	0.82%	1.11%	0.56%	1.20%	0.66%	7.78%	5.71%	14.87%	5.72%
Public_Health	Uni-modal	2.18251	0.96282	1.78257	0.88540	1.96002	0.91906	2.20737	0.92223	1.58171	0.88266	2.02351	0.98018
	Time-MMD	2.06260	0.93598	1.77603	0.91859	1.66797	0.84250	1.97107	0.86188	1.52698	0.85382	1.78536	0.92827
	Cross-MoE	1.76146	0.89517	1.48646	0.82041	1.62930	0.83684	1.75704	0.85785	1.44565	0.83371	1.79700	0.91563
	Promotion	14.60%	4.36%	16.30%	10.69%	2.32%	0.67%	10.86%	0.47%	5.33%	2.36%	-0.65%	1.36%
Security	Uni-modal	86.12715	5.21692	108.37794	5.07056	93.81701	5.61057	97.79772	5.76584	79.60748	4.77460	111.46080	5.18332
	Time-MMD	85.87784	5.10847	83.67643	4.99882	82.45282	4.96077	83.05544	4.93340	85.73862	5.32512	114.07553	5.41010
	Cross-MoE	79.66166	4.63439	79.57533	4.76940	80.60530	4.85269	82.22611	4.76342	82.92518	5.06522	108.79348	5.01120
	Promotion	7.24%	9.28%	4.90%	4.59%	2.24%	2.18%	1.00%	3.45%	3.28%	4.88%	4.63%	7.37%
SocialGood	Uni-modal	1.06997	0.44779	1.11839	0.53028	1.20396	0.45166	1.28119	0.47893	0.98422	0.52514	1.24355	0.59567
	Time-MMD	1.09304	0.50062	1.16969	0.51189	1.14853	0.44240	1.18187	0.41490	1.02886	0.59481	1.21705	0.61340
	Cross-MoE	1.03877	0.41481	1.05332	0.47823	1.08589	0.40513	1.21241	0.41680	0.96424	0.50388	1.05756	0.54234
	Promotion	4.96%	17.14%	9.95%	6.58%	5.45%	8.43%	-2.58%	-0.46%	6.28%	15.29%	13.10%	11.59%
Traffic	Uni-modal	0.18489	0.19625	0.24509	0.32775	0.18713	0.20294	0.20429	0.20864	0.29084	0.44693	0.23636	0.32482
	Time-MMD	0.20315	0.22273	0.23354	0.26295	0.19286	0.20720	0.19543	0.20209	0.25720	0.39620	0.21626	0.30752
	Cross-MoE	0.18767	0.19644	0.22620	0.26165	0.18105	0.18660	0.20804	0.21788	0.25375	0.39697	0.20410	0.29270
	Promotion	7.62%	11.80%	3.15%	0.50%	6.12%	9.94%	-6.45%	-7.81%	1.34%	-0.19%	5.63%	4.82%
Average Promotion		7.96%	7.49%	9.41%	6.46%	7.73%	6.25%	3.49%	1.00%	15.07%	8.40%	9.00%	6.72%

2024), Timesnet(Wu et al., 2022), Reformer(Kitaev et al.) and TimeLLM(Jin et al., 2024) as the time series models within our framework. We select the best-performing MoE network with a specific number of experts as our final result. In the prediction tasks, our method achieves an average 7.27% reduction in MSE compared to the Time-MMD framework, outperforming Time-MMD on most datasets. The evaluation metrics used are MSE and MAE. The specific results are shown in Table 1.

The specific calculation formula of the "Promotion" value is:

$$\text{Promotion} = \left(\frac{MSE_{\text{Time-MMD}} - MSE_{\text{Cross-MoE}}}{MSE_{\text{Time-MMD}}}\right)$$

5.2.1 Cross-Dataset Performance Comparison

Experimental results demonstrate that the Cross-MoE framework exhibits significant advantages across most datasets. Specifically:

High Improvement Areas: The framework shows the most notable performance gains on the Economy and Energy datasets, with average MSE improvements of 21.10% and 9.73%. These two datasets generally exhibit softer fluctuations, but with occasional sharp increases or decreases. Cross-MoE effectively aligns the information in text with these sharp changes, which leads to such significant improvements.

Stable Performance Areas: The Algriculture, Public Health, Security and Social good datasets achieve improvements of 8.31%, 5.72%, 4.59%, and 7.98%, respectively. On these datasets, the performance of Cross-MoE is relatively stable, and these datasets do not show strong common characteristics; the ratio of high to low-frequency components is fairly balanced.

Challenging Areas: On the Climate, Environment and Traffic datasets, the improvement was rel-

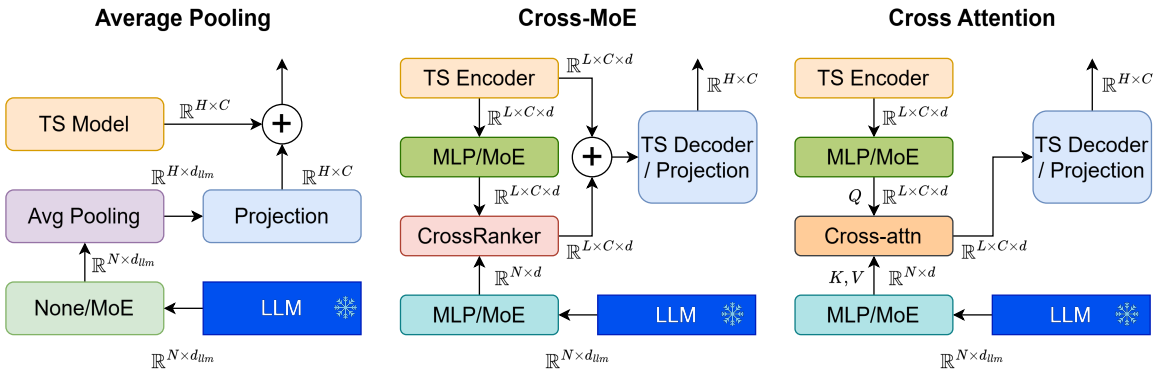

Figure 3: The network architecture of various Fusion methods.

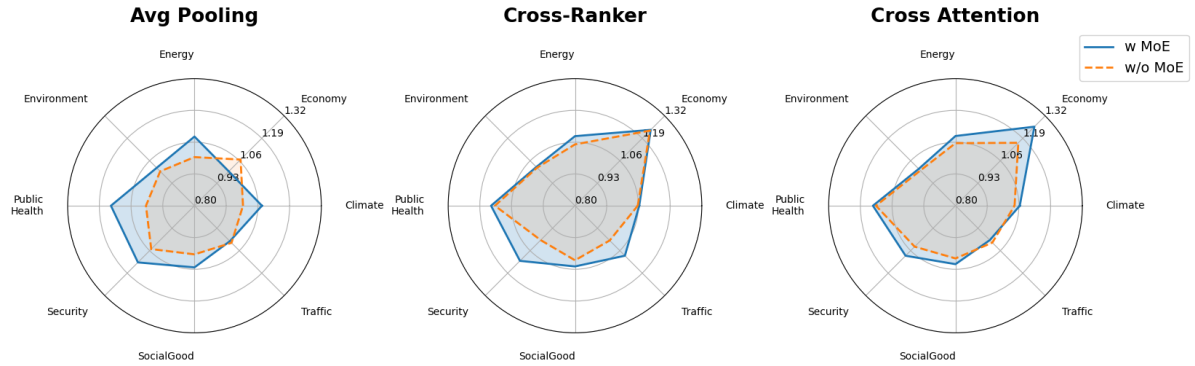

Figure 4: The normalized performance of the three fusion methods across all 9 datasets, where 'w' and 'w/o' denote 'with' and 'without', respectively. The longer radius indicates better performance differences resulting from the introduction of MoE.

atively modest (an average of 3%). These datasets contain rich high-frequency components in the time series, with values fluctuating sharply over time. This may indirectly suggest that Cross-MoE's ability to extract short-term, high-frequency temporal information from the text is insufficient.

5.3 Ablation Study

All ablation experiments were conducted using TimeXer as the TS model. Results are shown in Figure 4.

As shown in Figure 3, three distinct modality fusion methods were evaluated. In this section, extensive ablation experiments were conducted to demonstrate the performance improvements brought by Cross-MoE (Cross-Ranker combined with MoE), with comparisons to traditional Cross Attention and Time-MMD (Avg Pooling without MoE).

First, Avg Pooling averages the features of different tokens along the temporal dimension and replicates them C times to match the output shape of the temporal model. In this fusion method, MoE intervenes before the pooling operation. The second fusion method, Cross-Ranker, is our proposed approach. It calculates the similarity between the temporal model encoder output and the text embeddings, then ranks and selects the top $C \cdot L$ tokens based on the scores. MoE intervenes before the temporal features enter the Cross-Ranker. Lastly, Cross Attention, a popular modality fusion technique, uses the output of the temporal model encoder as the queries and the text features as the keys and values. The output shapes of all three fusion methods match exactly with the encoder output of the temporal model, allowing them to seamlessly integrate into the original model.

5.3.1 Performance Analysis

The introduction of MoE significantly improves performance across all fusion methods. On average, MSE reduction rates increased by over 50%, validating MoE's effectiveness in enhancing multimodal interactions. On the other hand, regardless of whether MoE is included, Cross-Ranker consistently outperforms Cross Attention on all

Table 2: Breakdown and derivation of the computational overhead introduced by different fusion strategies.

(a) Cross-Attention (Full)

Component	FLOPs Formula	Calculation Process	Subtotal (GFLOPs)
Q Projection	$B \times L \times d^2$	$32 \times 24 \times 512^2$	0.201
K Projection	$B \times N \times d^2$	$32 \times 512 \times 512 \times 512$	4.295
V Projection	$B \times N \times d^2$	$32 \times 512 \times 512 \times 512$	4.295
QK, AV Matrix	$B \times N \times L \times d$	$2 \times 32 \times 512 \times 24 \times 512$	0.402
FFN Layer	$8 \times B \times L \times d^2$	$8 \times 32 \times 24 \times 512^2$	1.611
Theoretical Total			**10.805**

(b) Cross-MoE (Ours)

Component	FLOPs Formula	Calculation Process	Subtotal (GFLOPs)
TX-MoE	$B \times N \times d_{llm} \times (M_{tx} + d)$	$32 \times 512 \times 768 \times (8 + 512)$	6.543
TS-MoE	$B \times L \times d \times (M_{ts} + d)$	$32 \times 24 \times 512 \times (8 + 512)$	0.204
Cross-Ranker	$B \times N \times L \times d + B \times N$	$32 \times 512 \times 24 \times 512 + 32 \times 512$	0.201
Theoretical Total			**6.948**

(c) Avg Pooling (Time-MMD)

Component	FLOPs Formula	Calculation Process	Subtotal (GFLOPs)
Feature Projection	$B \times N \times d_{llm} \times d_{llm}/8$	$32 \times 512 \times 768 \times 768/8$	1.208
Horizon Projection	$B \times N \times H \times d_{llm}/8$	$32 \times 512 \times 12 \times 768/8$	0.019
Theoretical Total			**1.227**

datasets, with an average performance advantage (e.g., 6.39% vs 5.46% for w MoE and 3.44% vs 2.83% for w/o MoE).

Specifically, the intervention of MoE has had contrasting effects on the Avg Pooling method across different datasets. On the Economy and Traffic datasets, the inclusion of MoE resulted in an increase in MSE by an average of 7.38% and 9.30%, respectively. However, for the other methods, MoE generally led to performance improvements. Compared to the case without MoE, Avg Pooling showed a 14.4% improvement on the Public Health dataset, Cross-Ranker improved by 12.8% on the Security dataset, and Cross Attention improved by 11.3% on the Economy dataset.

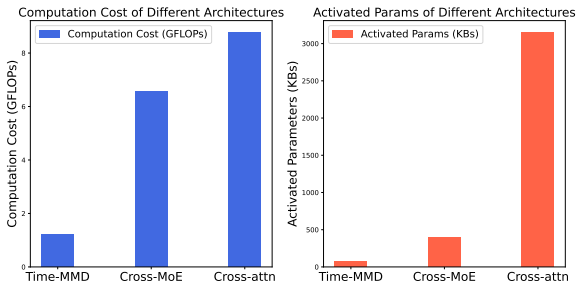

Figure 5: The overhead comparison of three fusion methods.

Additionally, the effectiveness of the three methods varies depending on the dataset. Cross-MoE and Cross Attention showed more significant improvements on the Economy and Public Health datasets. Notably, the Traffic dataset only saw substantial performance gains (a 7% improvement) when using Cross-MoE. In summary, Cross-Ranker and Cross Attention exhibit statistically comparable performance in most scenarios.

5.3.2 Overhead Analysis

We computed and summarized the additional overhead introduced by different fusion strategies under the conditions specified in Table 3. The theoretical calculation results are presented in Table 2.

Table 3: Model Parameters and Definitions

Symbol	Value	Definition
d_{llm}	768	LLM feature dimension
d	512	Temporal feature dimension
N	512	Text sequence length
L	24	Lookback window length
M_{ts}	8	TS-MoE experts
M_{tx}	8	TX-MoE experts
B	32	Batch size
H	12	Forecast horizon

As shown in Figure 5, compared to Cross Attention, Cross-MoE consumes 75% the computation and 12.5% of the memory usage.

The theoretical analysis aligns substantially with

the data presented in Figure 5, though minor discrepancies persist. These may stem from potential inaccuracies in the profiling tool or undisclosed optimizations within the operators. We ultimately elected to report the profiled results as the primary representation.

5.3.3 Hyperparameter Analysis

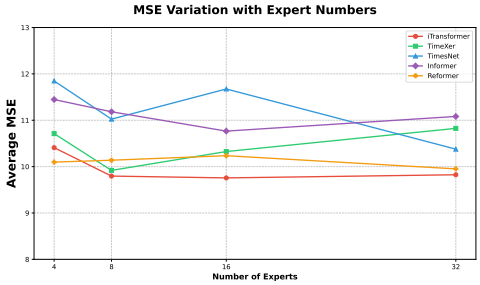

Figure 6: The MSE value across various expert numbers

We recorded the MSE values for three models with different numbers of experts (i.e. $M = \{4, 8, 16, 32\}$). As shown in Figure 6, the performance changes with varying numbers of experts are minimal. But the general trend is that as the number of experts increases, the MSE gradually decreases.

5.4 Visualization of MoE Router distribution

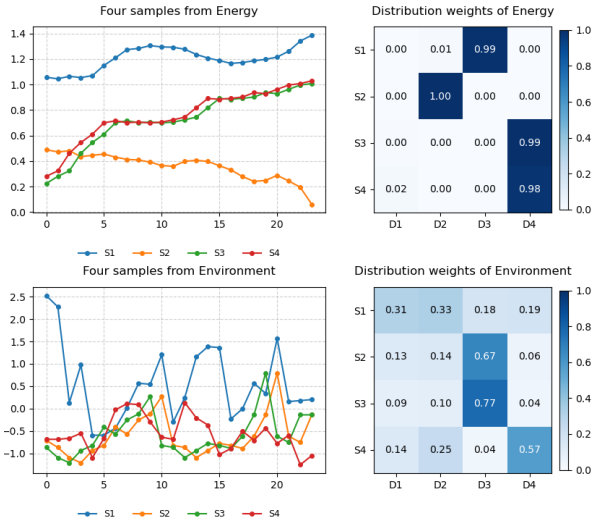

Figure 7: The distribution of experts output by Router in MoE.

We collected four distinct types of temporal intervals from both the Energy and Environment datasets, along with their corresponding Router output distributions. The results reveal that the Router in MoE effectively discriminate intervals with different patterns. As demonstrated in Figure 7, in the Energy dataset, samples S3 and S4 exhibit identical temporal patterns, while S1 and S2 demonstrate distinct pattern characteristics. While in the Environment dataset, the four samples exhibit less distinct pattern variations compared to those in the Energy dataset, resulting in a more balanced routing distribution.

6 Conclusion

A model-agnostic temporal-textual fusion framework is proposed in this paper. It aligns textual information with shifted temporal distributions through MoE networks, while employing a Cross-Ranker to filter irrelevant textual content. Compared with the current state-of-the-art, Cross-MoE achieves an 8.5% relative performance improvement while requiring only 75% of the computational overhead of conventional Cross Attention approaches to attain comparable predictive accuracy.

Limitations

The textual information incorporated in the current Time-MMD dataset consists of policies or news articles related to temporal data. Compared to direct descriptive annotations of temporal data, such textual descriptions lack explicit interpretability. Specifically, it remains unclear which key elements within the text contribute to specific impacts on the temporal data—whether they affect trends, periodic patterns, or noise components.

Additionally, in real-world scenarios, the sampling frequencies of temporal and textual data often mismatch. To address this, Time-MMD aligns textual content with temporal timestamps using a fixed time window, where each timestamp corresponds to all textual data within a predefined window preceding it. However, this alignment approach is relatively simplistic and risks introducing substantial irrelevant information and noise. Consequently, this may divert researchers' focus from exploring effective text-temporal fusion strategies to the necessity of filtering out redundant information.

Furthermore, the effectiveness of the MoE in large-scale temporal forecasting tasks remains unverified. The Cross-MoE framework has been validated exclusively on the small-scale Time-MMD dataset. Its scalability and validity in handling significantly larger text-temporal datasets require further investigation.

References

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31.

Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. 2017. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. 2023. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 7522–7529.

Daniel Hirschman and Elizabeth Popp Berman. 2014. Do economists make policies? on the political effects of economics. *Socio-economic review*, 12(4):779–811.

Furong Jia, Kevin Wang, Yixiang Zheng, Defu Cao, and Yan Liu. 2024. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21):23343–23351.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Farahnaz Pashaei Kamali, João AR Borges, Miranda PM Meuwissen, Imke JM de Boer, and Alfons GJM Oude Lansink. 2017. Sustainability assessment of agricultural systems: The validity of expert opinion and robustness of a multi-criteria analysis. *Agricultural systems*, 157:118–128.

Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. 2020. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4.

Kai Kim, Howard Tsai, Rajat Sen, Abhimanyu Das, Zihao Zhou, Abhishek Tanpure, Mathew Luo, and Rose Yu. 2024. Multi-modal forecaster: Jointly predicting time series and textual data. *arXiv preprint arXiv:2411.06735*.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

José Leal, Sarah Wordsworth, Rosa Legood, and Edward Blair. 2007. Eliciting expert opinion for economic models: an applied example. *Value in Health*, 10(3):195–203.

Raphael Lencucha, Nicole E Pal, Adriana Appau, Anne-Marie Thow, and Jeffrey Drope. 2020. Government policy and agricultural production: a scoping review to inform research and policy on healthy agricultural commodities. *Globalization and health*, 16(1):11.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2024a. Timecma: Towards llm-empowered time series forecasting via cross-modality alignment. *CoRR*.

Haoxin Liu, Shangqing Xu, Zhiyuan Zhao, Lingkai Kong, Harshavardhan Kamarthi, Aditya B. Sasanur, Megha Sharma, Jiaming Cui, Qingsong Wen, Chao Zhang, and B. Aditya Prakash. 2024b. Time-MMD: Multi-domain multimodal dataset for time series analysis. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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*.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024c. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*.

Donghao Luo and Xue Wang. 2024. Moderntcn: A modern pure convolution structure for general time series analysis. In *The twelfth international conference on learning representations*, pages 1–43.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *CoRR*, abs/2211.14730.

Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. 2020. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181.

Xiaoyu Tao, Tingyue Pan, Mingyue Cheng, and Yucong Luo. 2024. Hierarchical multimodal llms with semantic space alignment for enhanced time series classification. *arXiv preprint arXiv:2410.18686*.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. 2024. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pages 22419–22430. Curran Associates, Inc.

Zhijian Xu, Yuxuan Bian, Jianyuan Zhong, Xiangyu Wen, and Qiang Xu. 2024. Beyond trend and periodicity: Guiding time series forecasting with textual cues. *Preprint*, arXiv:2405.13522.

Jinsung Yoon, William R Zame, and Mihaela Van Der Schaar. 2018. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128.

Xiyuan Zhang, Diyan Teng, Ranak Roy Chowdhury, Shuheng Li, Dezhi Hong, Rajesh Gupta, and Jingbo Shang. 2024a. Unimts: Unified pre-training for motion time series. *Advances in Neural Information Processing Systems*, 37:107469–107493.

Xiyuan Zhang, Diyan Teng, Ranak Roy Chowdhury, Shuheng Li, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. 2024b. UniMTS: Unified pre-training for motion time series. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11106–11115. AAAI Press.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pages 27268–27286. PMLR.

A Appendix

A.1 Word Frequency Statistics

As shown in Table 4 tallied the top 10 words most frequently selected by Cross-Ranker across 9 different datasets where TimeXer serves as the TS model.

A.2 Visualization of forecasting results

We adopt TimeXer as the temporal model in our framework and conduct experiments on the Energy, Environment and Public Health datasets with a prediction horizon $H = 48$. As shown in Figure 8, the left side of the dashed line in the figure represents historical observations within the look-back window, while the right side displays predictions annotated with distinct colors corresponding to different fusion strategies. We observe that Cross-MoE achieves more accurate predictions compared to both Cross Attention and Avg Pooling. This improvement stems from its dual capability to filter irrelevant textual information while effectively aligning temporal patterns with text-based contextual cues.

In the Energy dataset characterized by frequent abrupt drops and surges, the model successfully leverages critical event indicators from text when such information is available. Conversely, the Environment dataset exhibits rich high-frequency components with relatively stable patterns, where all three fusion methods demonstrate limited effectiveness in extracting actionable insights from text. The Public Health dataset shares similarities with Energy in requiring textual guidance for trend interpretation, though it differs in exhibiting less periodic information that necessitates stronger reliance on external textual cues for forecasting.

A.3 Detailed Statistics of different fusion methods

Fig 9 demonstrates the detailed forecasting MSE results of Figure 4.

Table 4: The rankings of the top 10 most frequently occurring words across different datasets.

Dataset Name	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Agriculture	prompt	2016	according	agricultural	predictions	start	per	source	follows	##w
Climate	ed	based	48	near	across	conditions	##dent			
Economy	trade	predictions	source	future	restrictions	org	japan	states	gov	reduce
Energy	source	predictions	oil	prices	gov	crude	natural	price	2021	rising
Environment	air	quality	source	new	york	gov	epa	environmental	state	water
Public Health	based	information	predictions	influenza	##bi	source	ni	infection	various	states
Security	follows	help	##a	federal	source	state	key	5	##li	earthquakes
Social Good	nie	p	gov	6	unemployed	information	12	source	unable	##ls
Traffic	source	volume	gov	united	tr	com	high	data	vehicle	michigan

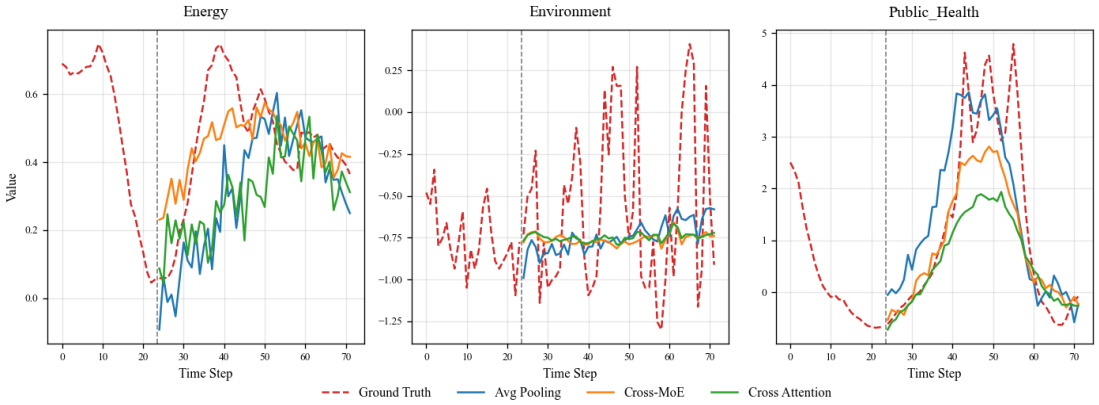

Figure 8: Visualization of the forecasting results.

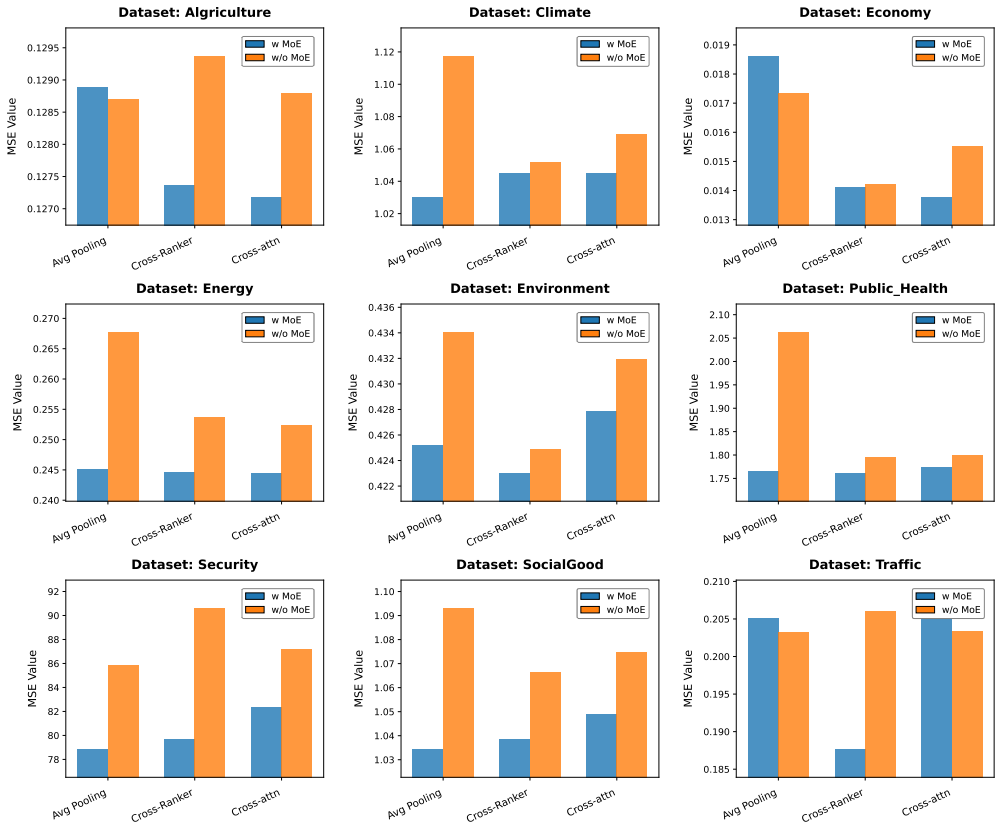

Figure 9: Detailed performance of different fusion methods.