

MiNiformer: Enhance Vanilla Transformer with Mixer-Adapter for Long-term Traffic Forecasting

Shaojun E^{1,2,5}[0009-0004-3766-2988], Wenjuan Han^{1,2,3,4}[0000-0002-2327-0842], Zhiwei zhang^{1,2}[0009-0004-5343-8230] and Jinan Xu^{1,2}[0000-0003-0170-626X] (✉)

¹ Beijing Key Lab of Traffic Data Analysis and Mining

² Beijing Jiaotong University, Beijing, China

³ China Railway Design Corporation, Tianjin, China

⁴ National Engineering Research Center for Digital Construction and Evaluation of Urban Rail Transit, Tianjin, China

⁵ Global Tone Communication Technology Co., Ltd., Beijing, China
{23140102, wjhan, 20271259, jaxu}@bjtu.edu.cn

Abstract. Recently, there’s been a surge in scholarly interest in traffic forecasting. Most of the efforts have been concentrated on short-term forecasting, and have yielded promising results. Long-term forecasting, though more practical, presents two challenges. First, existing approaches primarily capture dependencies and correlations within short-term historical data. Their performance drops when handling long-term spatio-temporal forecasting, indicating limited scalability. Second, most approaches tend to emphasize temporal information, often at the expense of neglecting important spatial geographic information. In response to these two challenges, we propose our transformer-based traffic forecasting approach, MiNiformer, featuring the Spatial Feature Extractor – Mixer Adapter as a crucial element. MiNiformer excels in extracting and integrating spatial features, leading to impressive results. Experiments show that MiNiformer, by leveraging spatial information and long-term dependencies, showcases robust long-term feature extraction capabilities and performs exceptionally well in both short-term and long-term scenarios.

Keywords: Spatio-temporal, traffic forecasting, Transformer, Mixer-adapter

1 Introduction

In the realm of Intelligent Traffic Systems (ITS), traffic forecasting occupies a critical role as a spatio-temporal data mining task[1]. It entails leveraging algorithms to unearth patterns within historical traffic data, subsequently enabling forecasts of future traffic flow. Traffic forecasting directly impacts people’s daily lives, with accurate and efficient models empowering informed travel decisions and lifestyle adjustments. Conventional approaches aim to extract the underlying distribution and salient features of the historical data, subsequently enabling the forecasting of future traffic patterns within a

✉ Corresponding author

specified time window. The forecasting horizon determines the classification of the problem into short-term or long-term forecasting [2][3][4]. The forecasting length is often divided by a one-hour boundary, with less than one hour for short-term forecasting and more than one hour for long-term forecasting [5]. In practical applications, the duration of the forecasting interval offers valuable insights for traffic management strategies. Longer forecasting horizons tend to be more advantageous, particularly for traffic flow data encompassing the next few hours. This extended time frame facilitates user path planning and optimizes traffic resource allocation. However, long-term forecasting presents two key challenges despite its advantages. Firstly, most current methods excel at capturing dependencies and correlations within short-term historical data. This focus, however, leads to performance degradation when applied to long-term spatio-temporal forecasting, highlighting limitations in scalability. Secondly, a prevalent bias exists towards temporal information, often neglecting the critical role of spatial geographic features. This omission hinders the ability of existing models to fully capture the complex interplay between space and time in traffic flow dynamics. Conventional approaches often rely on graph-based methods, such as spatio-temporal Graph Neural Networks (STGNNs) [6]. These approaches model historical traffic flow data by treating traffic data collection sensors as nodes within a Graph Neural Network (GNN) framework [7]. However, graph-based methods require the prioric definition of the graph structure’s typology, which significantly hinders the model’s transferability to road networks with distinct typologies. This pre-definition restricts the model’s focus solely on the training data’s typology, leading to catastrophic performance drops when evaluated on road networks with different typologies. In recent years, Transformer-based models[8][9][10] have emerged as an alternative and adopt a different way from graph-based methods, eschewing spatial typologies in favor of focusing on the temporal characteristics of time series data. These models have demonstrated promising performance, fueling a surge in scholarly interest. Unfortunately, they disregard the crucial role of spatial information in real-world traffic forecasting. To compensate for this omission, some efforts have incorporated spatial information through implicit spatial embedding modules [10][11]. The implicit nature of these designs hinders interpretability, as it remains unclear which specific spatio-temporal features contribute most significantly to enhanced prediction performance.

To address these challenges, we present an intuitive approach, the Mixer-Adapter Noise-Embedding Interfaced Transformer, denoted as MiNiformer. The cornerstone of MiNiformer is the Mixer-Adapter, a plug-and-play module adept at extracting and representing spatial geographic features of the traffic signal matrix \mathbf{I} . In addition, traffic flow data often exhibits “spike noise”, characterized by surges in flow at specific instances. To effectively model such noise, we incorporate a learnable noise embedding module, Noise Embedding. In summary, the contributions can be summarized as follows:

- We first propose a flexible and learnable adapter for spatial information - Mixer Adapter, which effectively compresses and extracts spatial information through a specially designed feature extraction module. It can explicitly extract spatial features

and integrate them with temporal features. This makes the model highly spatially sensitive, thereby improving forecasting accuracy and robustness.

- We integrate a learnable noise embedding module with the vanilla transformer. This module has the ability to comprehend and learn from the noise. The term “noise” here refers to the random or inconsistent part of the data that may not contribute directly to the output but may have an underlying pattern or structure that could be beneficial to learn.
- To verify MiNiformer’s effectiveness in long-term scenarios as well as short-term scenarios, we conducted extensive experiments on MiNiformer based on PEMS04 and PEMS08 to demonstrate its superiority over existing methods.

2 Relate work

2.1 Traffic Forecasting Task

The intelligent transportation system is a crucial component of smart mobility, and traffic forecasting is a significant aspect of intelligent transportation systems[12]. Accurate and reliable forecasting can alleviate congestion, conserve resources, guide people’s travel, reduce traffic accidents, and mitigate energy waste. Broadly speaking, traffic data falls under spatio-temporal data, making traffic forecasting a spatio-temporal data mining problem. Specifically, the task involves modeling based on historical data, where the model fits the characteristics of the historical data and then predicts data for a certain future period. The evaluation criteria are the accuracy of the forecasting for future data and the associated error loss. The better the accuracy and the lower the error, the more effective the modeling for the problem. Depending on the forecasting horizon, traffic forecasting can be categorized into short-term and long-term forecasts, typically divided at one-hour intervals. forecasting cycles shorter than one hour are considered short-term forecasts, while those longer than one hour are defined as long-term forecasts[2][3][4].

2.2 Graph-based Approaches

The fundamental challenge of the traffic forecasting task is the effective modeling of the responsible and dynamic spatio-temporal dependencies of traffic data[8]. Many methods, such as ARIMA[13] and SVM[14], only consider temporal information, but spatial dependency still plays a significant role in the problem of traffic forecasting. Therefore, from the perspective of extracting spatial features, we divide the methods into implicit spatial information extraction methods and explicit methods. There has been a lot of work on explicit methods of spatial features, with Convolutional Neural Networks (CNNs)[15] commonly used in the early stages for grid-based traffic data to capture spatial dependencies. Later, graph-based methods[16] have been proven to be more suitable for modeling the underlying graph structure of traffic data. However, methods based on GNNs often require a predefined structure, which is fixed and can

not be modifiable. It can be considered that once the topology of the road network changes, the prediction performance drops.

2.3 Transformer-based Approaches

Methods based on the Transformer often focus on the embedding encoding of input data, thus implicitly modeling the spatio-temporal data. Through a large amount of time series data and specially designed embedding modules, an implicit spatial representation is learned.

Due to the self-attention mechanism's advantages of parallelism, designing model architectures using self-attention mechanisms is highly attractive. Therefore, in 2017, the Google research team proposed a model entirely based on the self-attention mechanism: Transformer[17]. Initially proposed as a model for processing text data, the Transformer has gradually been applied to fields such as computer vision, reinforcement learning, and spatio-temporal data mining.[36][37]

Temporal-spatial models based on the Transformer architecture are gradually gaining attention from researchers, with numerous studies dedicated to this area in recent years[10][18]. ODformer[18] introduces a new attention mechanism on top of the Transformer architecture, aimed at capturing the special spatial dependencies between OD (Origin-Destination) pairs with the same origin (destination). This mechanism improves the model's predictive capabilities across various application scenarios by identifying and enhancing the correlations between OD pairs. PDFormer incorporates dynamic positional encoding into the transformer, enabling the model to better capture long-term dependencies in time series data. With an improved attention mechanism, it becomes more sensitive to positional information, thus achieving better results in tasks like sequence forecasting. STAEformer[10] introduces an implicit, learnable temporal-spatial embedding module at the embedding level to sensitize the model to temporal-spatial data, achieving commendable results through serial time-space attention modules. In summary, Transformer-based traffic forecasting methods have been receiving increasing attention in recent years, and a wealth of experiments has demonstrated that Transformer-based architectures can more effectively handle temporal-spatial related data. However, it cannot provide a reasonable and reliable reason in terms of interpretability. Therefore, how to explicitly and efficiently flexibly extract spatial information features is a direction worth paying attention to.

3 Preliminaries

Traffic Signal Matrix The traffic signal matrix is an adjacency matrix, that shows the spatial topological structure of the road network, with elements at each position representing both the connectivity and distance between two locations (namely, the location of the sensor). It is determined by an undirected graph $G = \{V, E, W\}$, where $V = \{V_1, V_2, \dots, V_n\}$ are nodes, representing n locations. E is the set of undirected edges, representing the connected nodes V_i and V_j . W is the matrix of two nodes. $G \in R^{n \times n}$ stores the distance of connected nodes, G is specifically determined by Equation (1).

$$G_{ij} = \begin{cases} w_{ij} & \text{if } E_{ij} \neq 0 \\ 0 & \text{else} \end{cases} \quad (1)$$

Traffic Forecasting We use X_t to denote the traffic signal matrix for the time step t . Starting from the sequence $X = (X_{t-P+1}, X_{t-P+2}, \dots, X_t)$ of the past P time steps, predict the future traffic signal matrix sequence $Y = (X_{t+1}, X_{t+2}, \dots, X_{t+Q})$ for the next Q time steps. This paper focuses on long-term traffic forecasting, hence both the input and output time spans exceed one hour.

4 MiNiformer

The core idea of our proposed MiNiformer, as shown in **Fig. 1**. Illustration of the architecture of MiNiformer. , lies in the explicit and efficient extraction of road network topology information. Besides designing a Mixer-adapter embedding module to compress spatial information, MiNiformer adopts a learnable noise module to simulate sudden data changes in the real world.

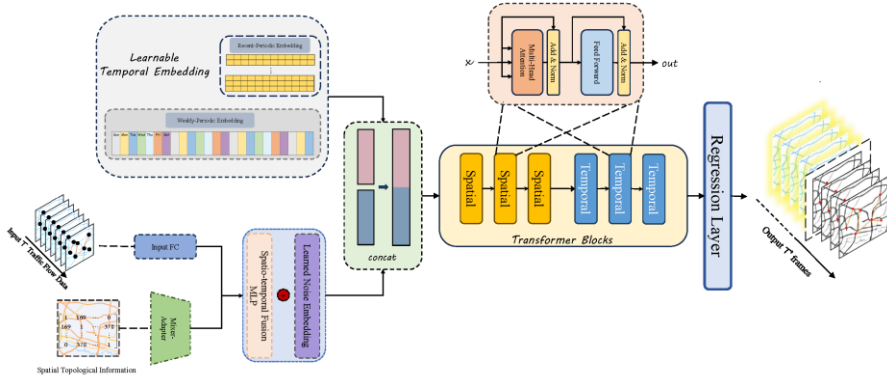


Fig. 1. Illustration of the architecture of MiNiformer.

4.1 Mixer-adapter.

How to effectively extract spatial information is a key issue in the field of spatio-temporal data mining. Prior research primarily explores two lines for incorporating spatial information: predefined network structures and learnable implicit embedding modules. Predefined network structures cannot learn the process of feature extraction from spatial information, and using learnable implicit embedding modules use spatial

information implicitly, reducing its interpretability. The Mixer Adapter, acting as a geographic information extractor, uses network structures as input and effectively extract spatial representations of network structures without the need for pre-definition. The input of network structures is independent of the time series data. This means that the Mixer-Adapter is an explicit, efficient, and learnable extraction method. Illustration of Mixer-Adapter is shown in **Fig. 2**

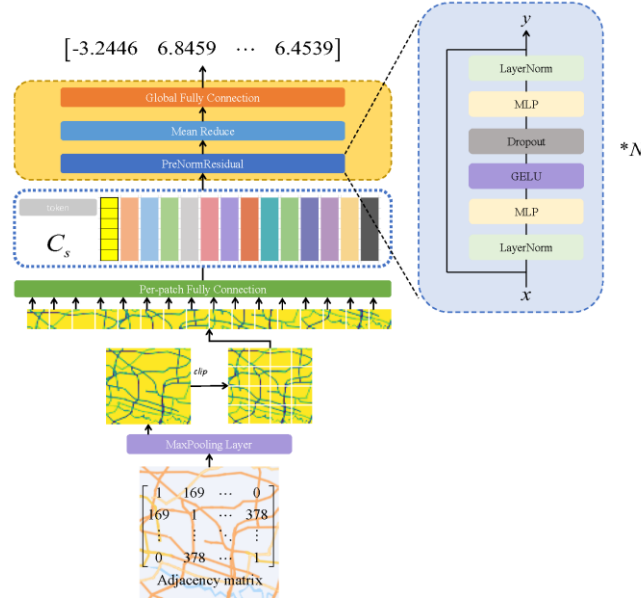


Fig. 2. Illustration of Mixer-Adapter. Following the pooling operation, the spatial topological structure is divided into multiple grids, which are then flattened into one-dimensional data and fed into the spatial extraction module. Eventually, they are processed into latent vectors containing spatial information.

Effectively and comprehensively modeling the dependency information of the network structure is not easy. We have fully considered how to compress and aggregate data from the original traffic signal matrix to enable the entire module to have better spatial feature learning and representation capabilities. For the traffic signal matrix, a max pooling operation is first adopted as follows:

$$X'_{i,j} = \max_{a=0}^{k-1} \max_{b=0}^{g-1} X_{s-i+a,t-j+b} \quad (2)$$

where $X'_{i,j}$ represents the value of the element at position (i,j) in the output feature map after the MaxPooling operation. $X_{s-i+a,t-j+b}$ indicates the value of the element within the pooling window in the input feature map X , corresponding to the position (i,j) in the output feature map, scaled by the strides s and t . k and g respectively represent the vertical and horizontal sizes of the pooling window. s and t are the vertical

and horizontal strides, determining the intervals at which the pooling window moves in the corresponding directions.

The matrix, after the pooling operation, condenses the location dependencies, connectivity, and distance information within the road network topology. This information, after going through two blocks of patch processing and matrix feature processing (See the following paragraphs), is compressed into latent vectors which preserve spatial topological information.

Patch processing Patch processing includes two steps: clipping and fully connected layers at the patch granularity. First, the $n \times n$ matrix is divided into non-overlapping blocks of size $p \times p$, with each block containing p^2 values. After flattening, the tensor dimension becomes p^2 , and these flattened tensors pass through a fully connected layer at the patch granularity, mapping the feature space dimensions to C_s . In this way, the information contained in each patch is mapped to an tensor, and these tensors together constitute the matrix where dimension is $[(n/p) * C_s]$.

Matrix Feature Processing. The matrix goes through three sub-layers, which are the residual block with the layer normalization, the mean reduce block, and the global fully connected layer. The residual block first applies layer normalization to the input, making the data distribution more uniform and enabling the model to have stable learning capabilities. The matrix, after layer normalization, is processed by an MLP that does not change the dimension, followed by an activation function and then another MLP layer that does not change the dimension too, and finally, it goes through layer normalization again. Moreover, to prevent overfitting, a dropout layer is added between the activation layer and the MLP. The output is obtained through a residual layer. The specific process can be represented by the following Equation. 3:

$$Out = x + LayerNorm(MLP(Dropout(\sigma(MLP(LayerNorm(x)))))) \quad (3)$$

After the residual connection operation, the matrix undergoes a dimension reduction through a mean normalization operation. By this point, the spatial information of the road network structure has been compressed into latent vectors. To better integrate with time series information, another MLP layer is introduced to unify the space of spatio-temporal data.

4.2 Embedding Layer

In Fig. 3, we categorize time series data into periodic series data with a daily periodicity, and short-term series data characterized by time step granularity obtained based on different sampling frequencies within a day. Through learnable embedding matrices — specifically, the weekly-periodic embedding $E_w \in R^{T * N_w * d_f}$ and the recent-periodic embedding $E_r \in R^{T * N_r * d_f}$, we learn the temporal features and information of these two trends, respectively. Here, $N_w = 7$ signifies that there are seven days in a week, while

$N_r = \frac{1}{f}$ depends on the sampling frequency f . Here, $N_r = 288$. $W^i \in R^T$ and $D^i \in R^T$

correspond to the periodic data and the recent data within the traffic time series for the interval $[X_{t-T+1}, X_{t-T+2}, \dots, X_t]$, which are used as indices to retrieve the corresponding

weekly-periodic embedding $E_i^w \in R^{T \times d_f}$ and recent-periodic embedding $E_i^r \in R^{T \times d_f}$ from their embedding dictionaries. By combining these two embedding modules, we obtain the periodicity embedding $E_{timestamps} \in R^{T \times N \times 2 \times d_f}$ for the traffic time series, thus facilitating an enhanced representation of temporal patterns within the data.

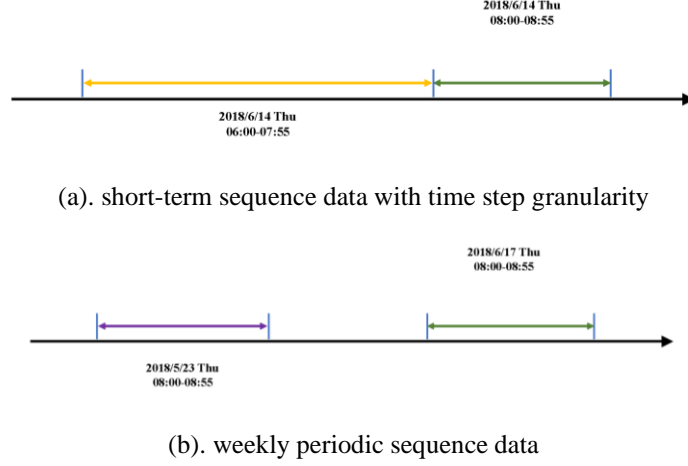


Fig. 3. Dividing time series data according to different trends: (a) short-term sequence information by sampling time units. (b) periodic information by day units.

4.3 Transformer and Regression Layer

As depicted in Figure 1, the encoder structure of the vanilla Transformer is applied to complex spatio-temporal traffic relationships: temporal encoder and spatial encoder. The temporal encoder focuses on capturing temporal information across temporal dimensions, and the spatial encoder captures spatial information. To ensure the efficiency of information capture, we adopt a “cascading structure” instead of a “cross-fusion” approach. Assume that the traffic latent variables in the latent space are represented by $Z \in \mathbb{R}^{T \times N \times d_h}$, where T denotes the number of time steps and N represents the number of spatial nodes (i.e., locations of sensors). The latent variables Z pass through a multi-head self-attention mechanism. Attention scores are obtained by multiplying with learnable parameter matrices, and the final computation result is obtained by multiplying with the values, the formula is as follows:

$$Z^t = \text{softmax}\left[\frac{(ZW_q^t)(ZW_k^t)}{\sqrt{d_h}}\right] \times (ZW_v^t) \quad (4)$$

where W_q^t , W_k^t and W_v^t are learnable parameter matrices. Through the computation of self-attention in the temporal dimension, we can extract the temporal features of the matrix. Similarly, applying a similar operation to the spatial dimension can extract spatial features:

$$Z^s = \text{softmax}\left[\frac{(Z^t W_q^s)(Z^t W_k^s)}{\sqrt{d_h}}\right] \times (Z W_v^s) \quad (5)$$

Finally, we leverage the output of the spatio-temporal transformer layers $Z^s \in \mathbb{R}^{T \times N \times d_h}$ to generate forecasting. The regression layer can be formulated as:

$$Y = \text{FullyConnection}(Z^s) \quad (6)$$

$Y \in \mathbb{R}^{T \times N \times 1}$ is the output forecasting result. By changing the dimensions through a fully connected neural network, the features are transformed into one dimension, thus obtaining the results to be predicted.

5 Experiments

5.1 Experimental Setting

Dataset. We selected two widely used datasets, PEMS04 and PEMS08. Both datasets are constructed by the Caltrans Performance Measurement System (PeMS) with a sampling frequency of once every 5 minutes, that is, 12 times per hour, 288 times per day. More detailed information is provided by **Table 1**.

Table 1. Summary of Datasets

Dataset	Sensors(N)	Time steps	Time Range	Time interval
PEMS04	307	16,992	01/2018-02/2018	5 min
PEMS08	170	17,856	07/2016-08/2016	5 min

Dataset Processing. We divide the dataset into training, validation, and test sets with a ratio of 6:2:2. Moreover, depending on the specific task, we sample data from different setup (24, 36, 48 steps) to predict future data of the same sampling granularity. The difference from previous tasks lies in our task’s focus on long-term traffic forecasting.

Baselines. We compare MiNiformer with the following baseline methods:

1. **History Average (HA):** Directly use the average of past time series data as the forecast value.
2. **Vector Auto-Regression (VAR) [19]:** The VAR method constructs models by treating each endogenous variable in the system as a function of the lagged values of all endogenous variables in the system, thereby avoiding the requirements of structured models.
3. **Deep Convolutional Recurrent Neural Network (DCRNN)[20]:** The DCRNN model is a deep learning model that combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), specifically designed to handle sequence data with spatial features and temporal dependencies, such as traffic flow forecasting.
4. **Graph WaveNet (GWNNet)[21]:** GWNNet is a neural network model for deep spatio-temporal graph modeling, which enhances the analysis and forecasting of

dynamic graph-structured data by combining graph convolution and recurrent units to capture complex spatial and temporal dependencies in graph data.

5. **Graph Multi-Attention Network (GMAN)[9]:** GMAN is a deep learning model that efficiently aggregates node features in a graph by employing multiple graph pooling strategies and attention mechanisms to fit spatio-temporal data.
6. **Adaptive Graph Convolutional Recurrent Network (AGCRN)[22]:** AGCRN captures complex spatio-temporal correlations in traffic sequences by introducing node-adaptive parameter learning modules and data-adaptive graph generation modules.
7. **Attention based Spatial-Temporal Graph Neural Network (ASTGNN)[23]:** ASTGNN is a traffic forecasting model that captures the local context of time series and the dynamic correlations of spatial data by combining self-attention mechanisms and dynamic graph convolution.
8. **Self-supervised Spatial Temporal Bottleneck Attentive Network (SSTBAN)[5]:** SSTBAN employs self-supervised learning and spatio-temporal bottleneck attention mechanisms, optimizing computational complexity and data utilization efficiency.
9. **STAEformer[10]:** STAEformer is a Transformer based model which has the spatial-temporal embedding E_a . It has achieved performance in short-term traffic flow forecast through different embedding vectors and attention mechanisms for different dimensions.

Run Setting. All experiments are conducted on a server equipped with an set of NVIDIA Force A6000 GPUs and 256GB of memory. The server’s operating system is Ubuntu 18.04. The code of MiNiformer is implemented entirely using Pytorch-2.1.0 and Python 3.9.7.

Evaluation Metrics. For the evaluation of the model, we selected three metrics: (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE). The formulas for these metrics are as follows:

- Mean Absolute Error (MAE): MAE assigns the same weight to all errors, meaning that both large and small errors have the same impact in the calculation.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (7)$$

- Mean Absolute Percentage Error (MAPE): MAPE can intuitively express the proportion of error relative to the actual values.

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (8)$$

- Root Mean Squared Error (RMSE): RMSE can better reflect the sensitivity of the model to outliers and the accuracy of forecasting.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (9)$$

5.2 Main Results

We comprehensively compared different time intervals with 8 models based on the PEMS04 and PEMS08 datasets. The results of these tasks can serve as a basis for long-term traffic forecasting capabilities. In **Table 1**, it can be seen that our MiNiformer generally achieved the best performance in the forecasting tasks of two datasets at different time steps. A more detailed observation are as follows.

1. The PEMS04 dataset contains 307 sampling points, and its complex data composition makes models more susceptible to the influence of discrete values. Despite the challenges, our MiNiformer achieved the best performance across all RMSE metrics. RMSE, by squaring the errors, amplifies larger errors and is more sensitive to significant errors. Our model demonstrates a good ability in sensitivity to outliers and forecasting accuracy.
2. On the PEMS08 dataset, MiNiformer consistently performed well in terms of MAE, which is the most intuitive metrics of the model’s ability to capture the trend. Thus, it can be considered that our model can effectively fit the target data.
3. Our model achieved the best performance on most of the 12 metrics. For the remaining metrics, it still performed second best. This fully demonstrates the robustness and reliability of our model.
4. Among models that explicitly utilize spatial information, SSTBAN has the best performance, but the use of spatial topological structure is not learnable, which means SSTBAN can not be transferred to other road networks with different spatial topological structures. On the contrary, extraction of spatial information for MiNiformer is learnable, making MiNiformer performs better.
5. STAEformer is a classic model that uses spatial information implicitly, reducing its interpretability. MiNiformer fully utilizes connectivity and distance cost, hence its performance is superior in most scenarios.

Table 2. Performance on PEMS04 and PEMS08.

Red marks the best performance; blue indicates second-best

Datasets	Metric	HA	VAR	DCRNN	GWNet	GMAN	AGCRN	SSTBAN	STAEformer	MiNiformer	
PEMS04	Step 24 (2 hours)	MAE	56.47	27.19	28.70	22.79	21.67	21.63	20.17	19.41	19.14
		RMSE	81.57	41.09	42.86	35.52	38.10	38.10	32.82	31.71	31.33
		MAPE(%)	45.49	21.42	21.23	16.04	17.78	17.78	14.43	12.68	12.54
	Step 36 (3 hours)	MAE	76.01	30.48	33.78	24.71	22.12	22.12	20.82	20.06	20.15
		RMSE	106.58	45.44	51.40	38.17	52.86	52.86	34.15	32.83	32.56
		MAPE(%)	68.84	24.15	27.10	17.67	16.43	16.43	14.83	12.94	13.17
	Step 48 (4 hours)	MAE	93.37	33.5	38.26	26.42	23.35	23.35	21.66	21.17	21.27
		RMSE	127.28	49.46	57.85	40.60	47.85	47.85	35.51	34.56	34.49
		MAPE(%)	94.62	27.28	33.73	18.99	17.98	17.98	15.90	13.78	13.82
P	Step 24	MAE	48.30	28.31	22.60	19.07	17.38	17.38	15.97	14.62	14.53

(2hours)	RMSE	69.72	44.47	33.34	29.47	34.29	34.29	26.32	25.96	25.01
	MAPE(%)	32.09	19.53	15.46	12.25	15.66	15.66	12.29	9.57	9.74
	MAE	65.99	31.70	25.82	21.76	17.21	17.21	16.84	15.27	15.22
Step 36 (3 hours)	RMSE	92.72	48.96	39.37	33.54	35.89	35.89	28.30	26.74	26.29
	MAPE(%)	46.64	22.56	18.53	13.68	16.33	16.33	12.20	10.05	10.35
	MAE	81.51	34.51	30.47	22.60	18.70	18.70	16.94	15.90	15.69
Step 48 (4 hours)	RMSE	111.85	52.14	45.64	34.20	48.54	48.54	28.82	27.89	27.31
	MAPE(%)	61.29	25.28	25.10	14.16	16.81	16.81	12.47	10.54	10.45

5.3 Experiment of Robustness

Fig 4. illustrates the performance and robustness of in comparison to base-lines across various time steps. Based on results reflected in Fig. 3. Dividing time scries data according to different trends: (a) short-term sequence information by sampling time units. (b) periodic information by day units., we can analyze the robustness of the models by checking if performance increases steadily as time steps increase. DCRNN and GMAN models show significant variations in MAE, RMSE, and MAPE metrics across different time steps. Although AGCRN and GWNet demonstrate more stability across different time steps, their performance is still inferior to MiNiformer. Compared to other base-lines, MiNiformer exhibits better robustness and stability, reflected in the more stable slope of its curve across forecasting over various time steps and achieving smaller errors. This demonstrates that the MiNiformer shows consistent effectiveness and exceptional performance in traffic forecasting tasks.

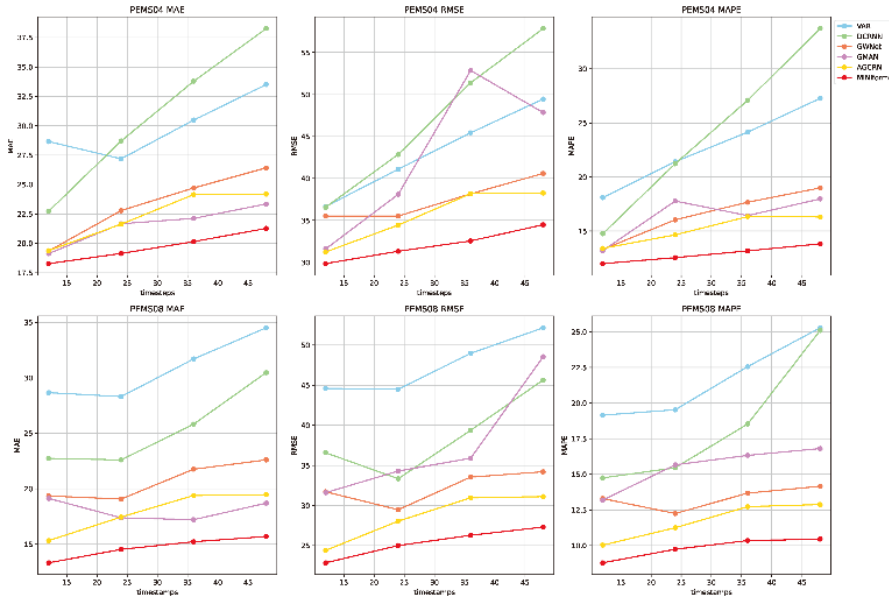


Fig. 4. Comparison of MiNiformer with baselines on various tasks is depicted in the top three graphs, showcasing the MAE, RMSE, and MAPE scores at different time steps on the PEMS04 dataset. Correspondingly, the three graphs on the bottom reflect the MAE, RMSE, and MAPE for the PEMS08 dataset.

5.4 Ablation Study

To further demonstrate the effectiveness of our two key modules (i.e., Mixer-Adapter and Noise Embedding), we conduct the ablation study on these two modules, comparing the complete MiNiformer with the version lacking the Mixer-Adapter and the version lacking both the Mixer-Adapter and Noise Embedding, in a step-by-step exploration of the effectiveness of each module. The results in **Table 3.** showcase the outcomes of our comparative experiments designed for different modules. It is evident that these two modules play a crucial role.

Table 3. Ablation Study On PEMS04 and PEMS08

	PEMS04-24			PEMS08-24		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Complete MiNiformer	19.14	31.33	12.54	14.53	25.01	9.74
W/o Mixer-Adapter	20.51	33.04	14.56	15.43	27.16	10.61
W/o Mixer-Adapter & Noise-embedding	24.99	40.25	16.76	17.43	29.78	11.39

6 Conclusion

We propose MiNiformer, that is capable of efficiently utilizing spatial information to address long-term traffic forecasting. Addressing the gap in past re-search, where explicit learning of spatial information was not possible, we de-signed a learning-capable spatial information capture module, the Mixer Adapter, allowing the model to consider spatial information while learning temporal data, thereby achieving explicit learning of spatial information and a paradigm for spatio-temporal data fusion processing. We have also integrated a noise learning module, significantly enhancing the model’s robustness. We conducted extensive experiments on two real-world datasets across three different forecasting intervals, demonstrating the superiority of MiNiformer.

7 Acknowledges

This work is supported by the Talent Fund of Beijing Jiaotong University (2023XKRC006) and the Pattern Recognition Center, WeChat AI, Tencent Inc.

8 References

1. M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3152–3168, 2020.
2. Z. Wang, X. Su, and Z. Ding, "Long-term traffic prediction based on lstm encoder-decoder architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2020.
3. Y. Hou, P. Edara, and C. Sun, "Traffic flow forecasting for urban work zones," *IEEE transactions on intelligent transportation systems*, vol. 16, no. 4, pp. 1761–1770, 2014.
4. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.
5. S. Guo, Y. Lin, L. Gong, C. Wang, Z. Zhou, Z. Shen, Y. Huang, and H. Wan, "Self-supervised spatial-temporal bottleneck attentive network for efficient long-term traffic forecasting," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1585–1596, IEEE, 2023.
6. G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng, "Spatio-temporal graph neural networks for predictive learning in urban computing: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
7. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.
8. J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 4365–4373, 2023.
9. C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1234–1241, 2020.
10. H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 4125–4129, 2023.
11. Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4454–4458, 2022.
12. M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent transportation systems*, vol. 21, no. 8, pp. 3152–3168, 2019.
13. G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neuro-computing*, vol. 50, pp. 159–175, 2003.
14. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.
15. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.
16. B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A gentle introduction to graph neural networks," *Distill*, vol. 6, no. 9, p. e33, 2021.

17. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
18. B. Huang, K. Ruan, W. Yu, J. Xiao, R. Xie, and J. Huang, "Odformer: spatial-temporal transformers for long sequence origin-destination matrix forecasting against cross application scenario," *Expert Systems with Applications*, vol. 222, p. 119835, 2023.
19. E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling financial time series with S-PLUS®*, pp. 385–429, 2006.
20. Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.
21. Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.
22. L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17804–17815, 2020.
23. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 922–929, 2019.
24. Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI'19*, p. 1907–1913, AAAI Press, 2019.
25. B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
26. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.
27. C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, pp. 914–921, 2020.
28. V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.
29. L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a General, Powerful, Scalable Graph Transformer," *Advances in Neural Information Processing Systems*, vol. 35, 2022.
30. L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, (New York, NY, USA), p. 547–555, Association for Computing Machinery, 2021.
31. K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.
32. P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73 – 101, 1964.
33. A. Einstein, "On the electrodynamics of moving bodies," 1905.
34. Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, dec 2022.

35. S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2021. 12
36. Yueting Yang, Xintong Zhang and Wenjuan Han, "Enhance Reasoning Ability of Visual-Language Models via Large Language Models", *arXiv preprint arXiv: <https://arxiv.org/abs/2305.13267>*
37. Zhao, Haozhe, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. "Mmicl: Empowering vision-language model with multi-modal in-context learning." *arXiv preprint arXiv:2309.07915* (2023).