Traffic & Transportation

Adaptive Denoising Spatio-Temporal Attention Network Fused With External Factors for Passenger Flow Prediction

Ruotong YANG[1], Lijuan LIU[2], Qinzhi LV[3]

Original Scientific Paper
Submitted: 13 Dec 2024
Accepted: 19 Mar 2025

[1] 2222031166@stu.xmut.edu.cn, School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China
[2] Corresponding author, ljliu@xmut.edu.cn, School of Computer and Information Engineering, Xiamen University of Technology; Fujian Key Laboratory of Pattern Recognition and Image Understanding, Xiamen, China
[3] 2222031141@stu.xmut.edu.cn, School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

Accurate passenger flow prediction is important in intelligent transportation systems (ITS). It is widely known that introducing external factors, such as weather and air quality data, can enhance the feature representation of passenger flow data, which has a certain positive impact on improving prediction performance. However, if there is no effective denoising, the more external factors introduced, the lower the prediction performance. Therefore, there are few studies that incorporate external factors into modelling. To this end, we propose an Adaptive Denoising Spatio-Temporal Attention Network (ADSTA-Net) that integrates external factors for passenger flow prediction. The core of the model is to fully consider the impact of external factors on passenger flow. Specifically, in the initial stage of ADSTA-Net, multiple external factors are combined with passenger flow. Then, the adaptive learning parameter matrix (ALPM) and fast Fourier transform (FFT) are applied to perform adaptive denoising for the fused features at different times and locations. Finally, a simplified Graph Multi-Attention Network (GMAN) with only-one layer ST-Attention block is used to learn global spatio-temporal dependencies. Extensive experiments are conducted on two real-world passenger flow datasets. The results demonstrate that ADSTA-Net has superior performance, particularly in making more accurate predictions under bad weather conditions.

KEYWORDS

passenger flow prediction; external factors; attention mechanism; adaptive denoising.

1. INTRODUCTION

Accurate passenger flow prediction can help solve some practical traffic problems, such as effectively easing road congestion, allocating resource, and so on [1–3]. Given the significant impact of historical passenger flow on prediction performance, most existing studies tend to map a correlation between historical and future flow. However, there are many different types of external factors, such as weather, holidays and air quality. Although their impact on prediction results may not be as significant as that of historical passenger flow, they still play a positive role in improving the model's performance. For example, *Figure 1* shows the passenger flow at Caitang Station of Xiamen bus rapid transit (BRT) during the same period but under different weather conditions over two consecutive weeks. It can be observed that during the same time period from 14:00 to 15:00 on different days, passenger flow exhibits different trends under two different weather conditions: rainy and non-rainy. Obviously, the changes in passenger flow are influenced by weather factors. Therefore, reasonably incorporating external factors is of great significance for improving the performance of models. To distinguish from external factors, we define traffic flow as internal factors.

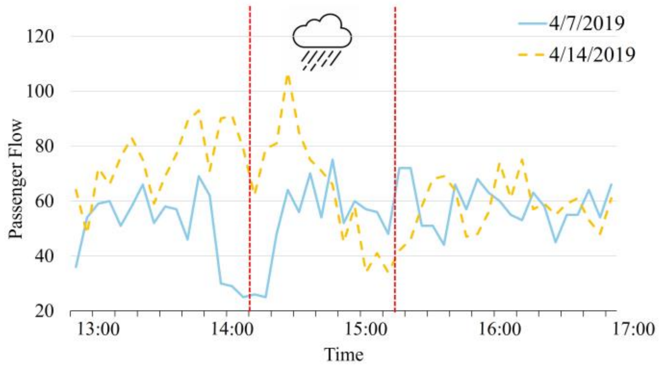


Figure 1 – Impact of rainfall on passenger flow at Caitang Station of Xiamen BRT

However, there are few studies that incorporate external factors into modelling. The possible reasons are summarised as follows: (1) It is often difficult to ensure that the obtained external factors have the same time granularity as the internal factors, which leads to the problem of time mismatch between internal and external factors. (2) Not all external factors have a positive impact on prediction performance. If too many useless external factors are directly introduced, it is easy to mix in a large amount of noise and reduce performance.

To better explore the impact of external factors on traffic flow prediction, we review the representative literature and list them in *Table 1*. In the literature, [4–6] use passenger flow, [7, 8] use vehicle flow, [5, 6, 9–11] use taxi trajectory, [9, 11] use bicycle trajectory, [11] uses bicycle flow. Their characteristics and existing problems are summarised as follows.

Table 1 – Summary of representative literature using external factors

Model	**External factor**				**Weather indicator**							**Structure**
	Weather	**Holiday**	**POIs**	**Air quality**	**Weather Conditions**	**Temperature**	**Wind**	**Rainfall**	**Visibility**	**Humidity**	**Air**	
MS-Net [5]	√	√	√		√	√	√	√	√			(a)
MCB+ATT [6]	√					√	√	√			√	(a)
SAE+RBF [8]	√				√	√	√		√	√		(a)
ST-ResNet [9]	√	√			√	√	√					(a)
STRN [10]	√	√	√		√	√	√					(b)
ResLSTM [4]	√			√		√	√			√	√	(b)
MVGCN [11]	√	√			√	√	√					(b)
DMSTSAF [7]	√				√	√	√					(c)

1) In terms of the diversity of external factors, the most widely used external factors are weather, holiday, points of interest (POIs) and air quality. For weather indicators, weather conditions, temperature and wind are the most commonly used in these models [5, 7, 8, 9–11]. Obviously, there is still a certain degree of subjectivity in deciding which external factors should be included in the model. In addition, to minimise noise, these models use no more than five weather indicators [4, 7, 10, 11] or a single categorical indicator of weather conditions, hoping to improve prediction performance by introducing as few weather indicators as possible [5, 7, 8, 9–11].
2) In terms of the structures of models, they are roughly divided into three types: (a) the first type shown in *Figure 2a* is to extract features from external factors through a feature network, while extracting spatio-temporal features from internal factors through another branch of a spatio-temporal feature extraction model (STFEM). Then, a fully connected layer (FC) is typically used to simply concatenate the features extracted from the two branches to construct a prediction model [5, 6, 8, 9]. (b) The second type shown in *Figure 2b* is to extract features from internal and external factors through two independent feature networks, respectively. Then, the features extracted from the two branches are fused together and the fused features are performed for further feature extraction through STFEM [4, 10, 11]. (c) The third one shown in *Figure*

2c fuses the internal and external factors in the initial stage and then feeds them into STFEM for spatio-temporal features extraction [7]. Compared with the first two types of models, the third type of model fully considers the fusion of internal and external factors in an earlier stage. However, this type of model is not commonly used. The possible reason is that the earlier the fusion occurs, the greater the negative impact of introduced noise on the effectiveness of feature extraction, which may reduce prediction performance.

3) In terms of adaptive denoising [12, 13], there is limited research on both integrating internal and external data and adaptively denoising the fused features in traffic flow prediction. For example, the typical model [7], which represents the third type, applies a spatial and temporal attention mechanism to adaptively integrate the internal and external factors, but it does not effectively denoise the fused features. Fortunately, traffic flow prediction is an important branch of time series prediction. Currently, there are some effective methods, such as the Bayesian [14], wavelet transform [15], fast Fourier transform (FFT) [16], autoencoder [17], semi-supervised learning [18] and unsupervised learning [19], which are successfully used for denoising time series data. It is worth mentioning that FFT converts signals from the time domain to the frequency domain and reduces noise by filtering out high-frequency signals. Its high accuracy, ease of implementation and wide applicability have made it widely used in time series prediction and achieved good results [16]. For example, Frequency improved Legendre Memory (FiLM) is proposed to use Fourier transform to reduce noise in six time series data [16]. However, it is only used to reduce noise of individual time series data and not for denoising fused features. Furthermore, to the best of our knowledge, there is relatively little research on adaptive denoising in the field of time series prediction, especially in passenger flow prediction. The related study [20] only uses an adaptive parameter matrix for weight matching to consider the impact of sudden traffic flows at different times and regions on predictions, rather than the adaptive parameter matrix for adaptive denoising of fused features.

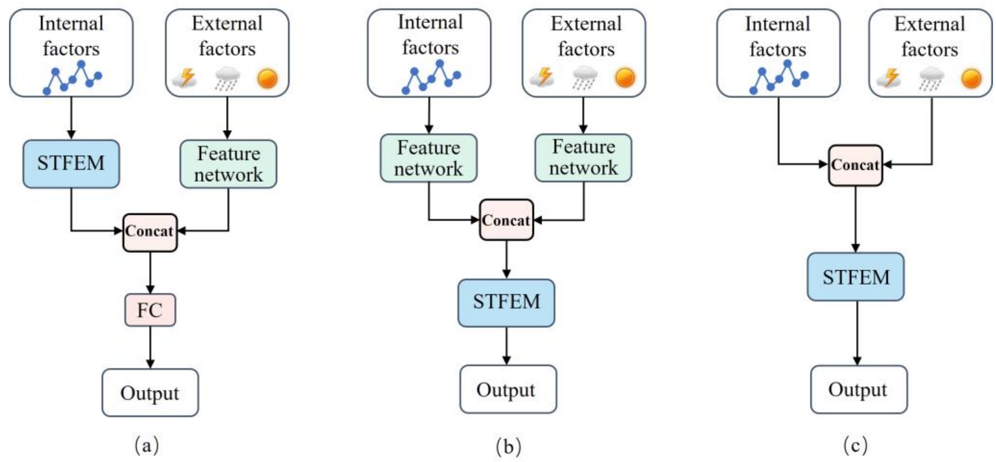


Figure 2 – Three types of models for internal and external factors fusing: (a) the first type; (b) the second type; (c) the third type

Therefore, the key to improving the performance of passenger flow prediction models that fuse external factors lies in how to fuse the internal and external factors and adaptively denoise the fused features. Given the shortcomings of the existing research, an Adaptive Denoising Spatio-Temporal Attention Network (ADSTA-Net) fused with external factors is proposed. The main contributions are summarised as follows:

1) A spatio-temporal information distillation module is designed for effective fusion of internal and external factors and adaptive denoising on the fused features. As opposed to previous studies, we integrate various external factors with historical passenger flow to obtain the fused features in the initial stage of ADSTA-Net, with the aim to enhance the feature representation of passenger flow data. Then, adaptive learning parameter matrix (ALPM) and FFT are used to adaptively denoise the fused features at different times and locations. In this way, the impact of external factors on passenger flow can be fully considered, which helps to improve the predictive performance.
2) To globally extract the useful features from the fused features, we use the Graph Multi-Attention Network (GMAN) [21] as the backbone and simplify it into a new network with a single-layer ST-Attention block to ensure the prediction performance while reducing the complexity of the model. Furthermore, layer normalization is added to the ST-Attention block and Wing loss is handpicked as the loss function, both of which can further improve the training capability of ADSTA-Net.

3) We evaluate the proposed ADSTA-Net on two real-world passenger flow datasets. The experimental results demonstrate that ADSTA-Net outperforms the existing models, particularly in making accurate predictions under the bad weather conditions.

This paper is organised as follows: Section 2 describes the details of ADSTA-Net. Section 3 analyses the experimental results. Finally, Section 4 summarises the paper and discusses the future research directions.

2. METHODOLOGY

2.1 Overall architecture

Figure 3a shows the architecture of ADSTA-Net, which is mainly composed of four parts: (1) the data fusion block (DFB) is designed to fuse external factors and passenger flow; (2) the adaptive denoising block (ADB) is designed to adaptively denoise the fused features, which mainly includes ALPM and FFT shown in *Figure 3b*; (3) the spatio-temporal embedding (STE) is devised to embed the dynamic spatio-temporal correlations for further feature extraction, which is shown in *Figure 3c*; (4) the spatio-temporal attention block (STAB) is proposed to extract global spatio-temporal dependencies for prediction, which is shown in *Figure 3d*.

In ADSTA-Net, the DFB first concatenates historical passenger flow X and external factors Ex to form the concatenated features X_{Ex}, and then performs the dimensional transformation on X_{Ex} mainly through a linear layer and FC to obtain $X_{Fusion}{}'$. The ADB is used to adaptively denoise $X_{Fusion}{}'$ through an ALPM and FFT in sequence, thereby obtaining X_{ADB}. X_{STE} is derived by adding spatial and temporal embedding. Then, X_{ADB} and X_{STE} are both used as the input to the STAB, and feature extraction is further performed through spatial and temporal attentions. After that, the global spatio-temporal dependencies X_{STAB} will be obtained through a gated fusion mechanism and layer normalization. Finally, the predicted passenger flow $\hat{Y}$ will be obtained through a FC. During the procedure, Wing loss [22] is handpicked as the loss function. The details of each block are shown in the following Section 2.2 and Section 2.3.

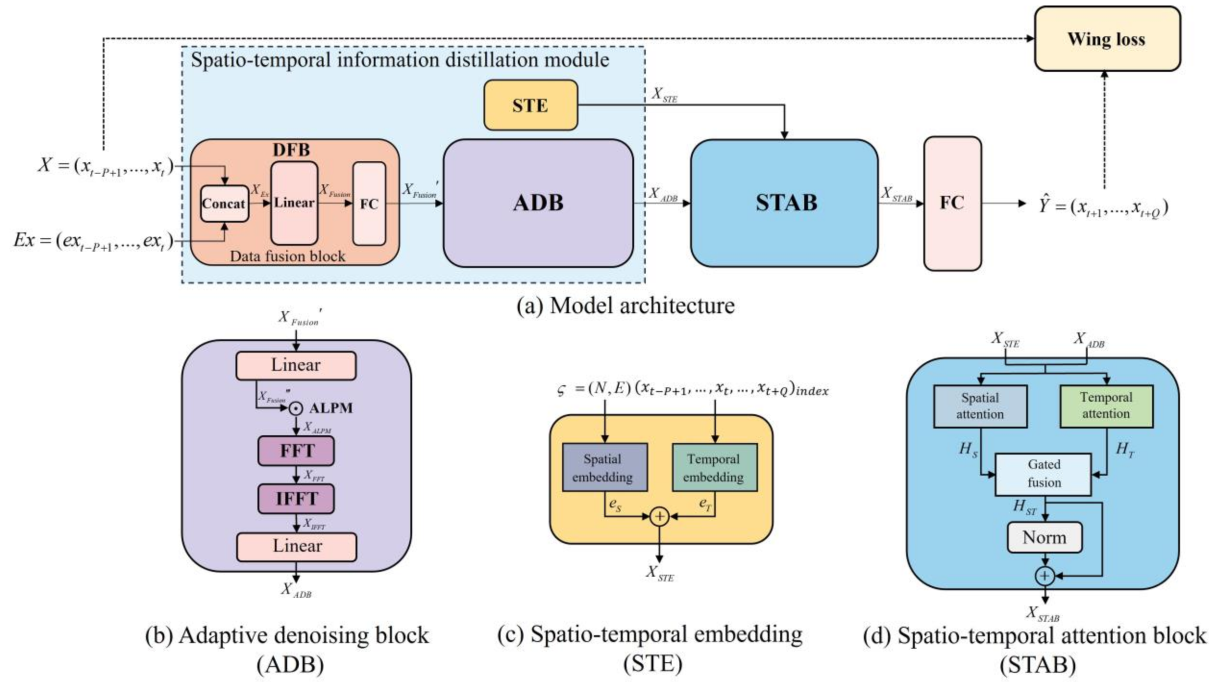


Figure 3 – The overall architecture of ADSTA-Net

2.2 Spatio-temporal information distillation module

As shown in *Figure 3a*, the spatio-temporal information distillation module mainly consists of three parts: the DFB, ADB and STE. As described in Section 1, X_{Ex} contains a lot of noise. Although X_{Ex} goes through a series of linear transformations to get $X_{Fusion}{}'$ in the DFB, $X_{Fusion}{}'$ still contains an amount of noise. Therefore, it is not appropriate to directly input $X_{Fusion}{}'$ into the STAB without any processing. To design a more competitive prediction model, we specifically design the ADB to adaptively denoise the features in $X_{Fusion}{}'$.

Data fusion block

There are two inputs $X \in \mathbb{R}^{P\times N}$ and $Ex \in \mathbb{R}^{P\times E}$ in the DFB. X and Ex are firstly concatenated as $X_{Ex} \in \mathbb{R}^{P\times(N+E)}$ based on *Equation 1*.

$$X_{Ex} = Concat(X, Ex) \tag{1}$$

Then, a linear layer is used to transform X_{Ex} to $X_{Fusion} \in \mathbb{R}^{P\times N}$ for feature extraction based on *Equation 2*:

$$X_{Fusion} = X_{Ex}W + b \tag{2}$$

where W and b are learnable parameters.

In order to effectively extract the global spatio-temporal dependencies in the subsequent STAB, we use FC to further transform the dimension of X_{Fusion} shown in *Equation 3*. FC uses the convolution with kernel size=1 for linear conversion and $ReLU$ as the activation function. After that, the dimension of X_{Fusion} is extended to $P\times N\times D$, and $X_{Fusion}{}' \in \mathbb{R}^{P\times N\times D}$ is obtained as the output of the DFB:

$$X_{Fusion}{}' = ReLU(X_{Fusion}W + b) \tag{3}$$

where W and b are learnable parameters. D is a new extended dimension of $X_{Fusion}{}'$.

Adaptive denoising block

$X_{Fusion}{}'$ will be continue to be input into the ADB for adaptive denoising. The dimension transformation in the ADB is shown in *Figure 4*. In the initial stage of the ADB, we perform dimension swapping on $X_{Fusion}{}'$ to obtain $X_{Fusion}{}'' \in \mathbb{R}^{N\times D\times P}$. In this way, we can easily perform FFT on $X_{Fusion}{}''$ since the temporal dimension P has been swapped to the last dimension. Considering that the dimension of P in $X_{Fusion}{}''$ is small, we use a linear layer to extend its dimension from P to M based on *Equation 4*, where $M>>P$. As a result, $X_{Fusion}{}''' \in \mathbb{R}^{N\times D\times M}$ is obtained:

$$X_{Fusion}{}''' = X_{Fusion}{}''W + b \tag{4}$$

where W and b are learnable parameters. M is an extended dimension of $X_{Fusion}{}'''$.

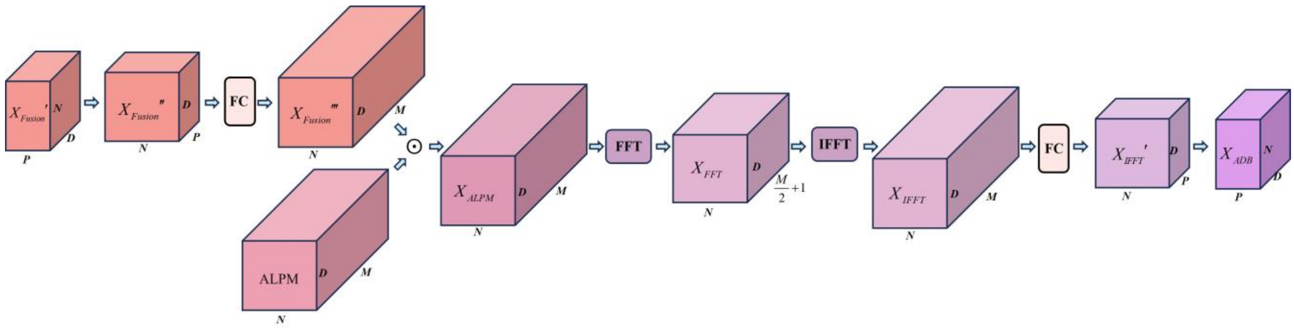


Figure 4 – Dimension transformation in the ADB

Then, we randomly initialise the weights in ALPM, whose dimension is consistent with that of $X_{Fusion}{}'''$. The weights in ALPM are all updated along with the other parameters. After the inner product between $X_{Fusion}{}'''$ and ALPM, we obtain $X_{ALPM} \in \mathbb{R}^{N\times D\times M}$ in *Equation 5*. In this way, we can adaptively assign higher weights to the key features and lower weights to the unimportant features for $X_{Fusion}{}'''$ through the ALPM:

$$X_{ALPM} = X_{Fusion}{}''' \odot ALPM \tag{5}$$

where $\odot$ represents the inner product.

Next, X_{ALPM} is sent to FFT for denoising. FFT is realised based on *Equation 6*, and $X_{FFT} \in \mathbb{R}^{N\times D\times(\frac{M}{2}+1)}$ is the obtained result. We select the first K sine wave signals with the lowest frequencies $argtop_K(X_{FFT})$ from

X_{FFT} as denoised features in the frequency domain signal, and then convert $argtop_K(X_{FFT})$ back into the time domain signal $X_{IFFT} \in \mathbb{R}^{N\times D\times M}$ via inverse fast Fourier transform (IFFT) shown in *Equation 7*. Compared to X_{ALPM}, X_{IFFT} represents the features after adaptive denosing by ALPM and FFT, which is more suitable for feature extraction in the subsequent STAB:

$$X_{FFT} = \sum_{m=0}^{M-1} X_{ALPM} \cdot e^{-j\frac{2\pi}{N}\omega n} \quad (6)$$

$$X_{IFFT} = \frac{1}{M}\sum_{\omega=0}^{M-1} [argtop_K(X_{FFT})] \cdot e^{j\frac{2\pi}{N}\omega n} \quad (7)$$

where X_{FFT} is the frequency domain signal, $e^{-j\frac{2\pi}{N}\omega n}$ is used to convert the signal from time domain to frequency domain, X_{IFFT} is the time domain signal and $e^{j\frac{2\pi}{N}\omega n}$ is used to convert the signal from frequency domain back to time domain.

Finally, we use a linear layer to change the dimension of X_{IFFT} from $N\times D\times M$ to $N\times D\times P$ based on *Equation 8*, obtaining $X_{IFFT}' \in \mathbb{R}^{N\times D\times P}$:

$$X_{IFFT}' = X_{IFFT}W + b \quad (8)$$

where W and b are learnable parameters.

Then we continue to perform dimension swapping on X_{IFFT}' and obtain the final output $X_{ADB} \in \mathbb{R}^{P\times N\times D}$ of the ADB. Through dimension swapping, we can make the dimension of X_{ADB} the same as that of X_{Fusion}'. Compared to X_{Fusion}', X_{ADB} is the result of adaptive denoising of X_{Fusion}'.

Spatio-temporal embedding

The STE has two inputs: the graph of the traffic network ς=(N, e) and a series of indexes of $P+Q$ time steps $(x_{t-P+1},\dots,x_t,x_{t+1},\dots,x_{t+Q})_{index}$, e.g. 2019-03-01 06:30:00. N and e represent the number of nodes and edges between nodes, respectively. P and Q are the step size of input and predicted flow, respectively.

The main function of the STE is to embed ς and $(x_{t-P+1},\dots,x_t,x_{t+1},\dots,x_{t+Q})_{index}$ to e^S and e^T, respectively. Dimension transformation in the STE is shown in *Figure 5*. The STE is divided into two parts: spatial embedding and temporal embedding, whose architecture is the same with the STE in GMAN [21]. Specifically, spatial embedding uses node2vec to generate a vector representation of N nodes in the traffic network, and its result is represented as $e^S \in \mathbb{R}^{N\times D}$. Temporal embedding uses one-hot encoding to encode $(x_{t-P+1},\dots,x_t)_{index}$ and $(x_{t+1},\dots,x_{t+Q})_{index}$ sequentially, and its result is represented as $e^T \in \mathbb{R}^{(P+Q)\times D}$. After that, we add e^S and e^T together and obtain the output $X_{STE} \in \mathbb{R}^{(P+Q)\times N\times D}$ of the STE.

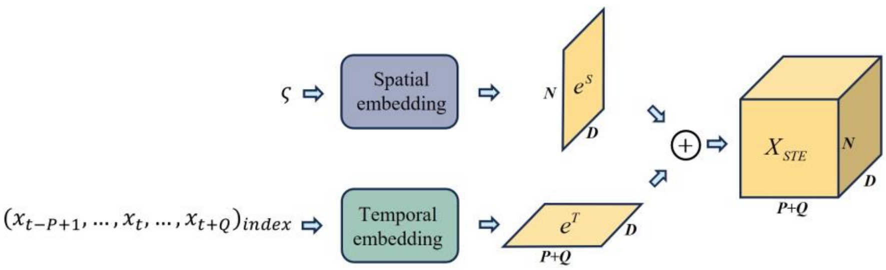


Figure 5 – Dimension transformation in the STE

2.3 Spatio-temporal attention block

As shown in *Figure 3a*, the STAB has two inputs: X_{STE} and X_{ADB}. The main function of the STAB is to further extract global spatio-temporal dependencies from both X_{STE} and X_{ADB} through a parallel attention mechanism. The dimension transformation in the STAB is shown in *Figure 6*. It contains spatial attention, temporal attention, gated fusion, layer normalization and residual connection. The processing is defined in *Equations 9–12*.

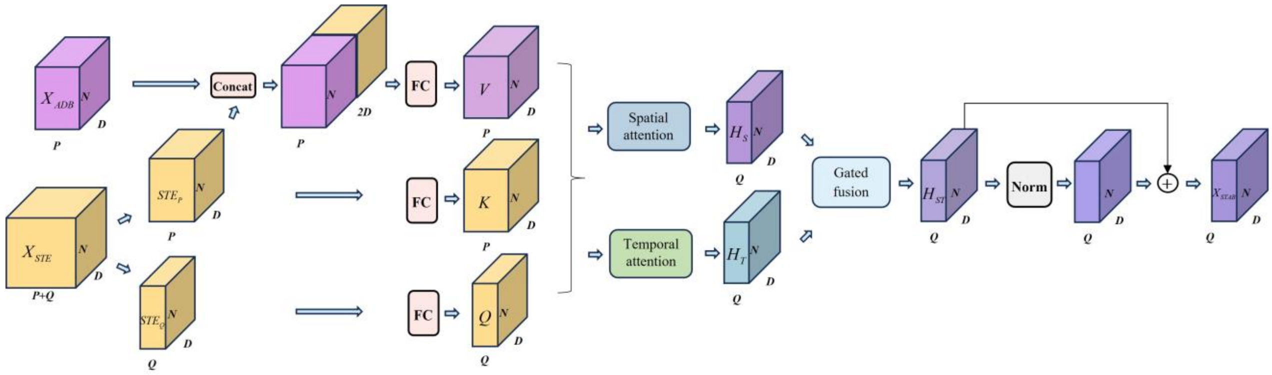


Figure 6 – Dimension transformation in the STAB

$$H_S = Attention_S(X_{ADB}, X_{STE}) \tag{9}$$

$$H_T = Attention_T(X_{ADB}, X_{STE}) \tag{10}$$

$$H_{ST} = Gated(H_S, H_T) \tag{11}$$

$$X_{STAB} = Norm(H_{ST}) + H_{ST} \tag{12}$$

In ADSTA-Net, the processing procedure of the STAB is mostly similar to that of the ST-Attention Block in GMAN [21]. However, the first difference is to add the layer normalization on $H_{ST} \in \mathbb{R}^{Q \times N \times D}$, which makes gradients smoother, trains faster and has better generalization ability [23] in ADSTA-Net. In addition, the query, key and value are also different. Specifically, in order to reduce the complexity of ADSTA-Net and comprehensively take into account the impact of X_{STE} on X_{ADB}, we only use single-layer ST-Attention Block in the STAB instead of the three-layer ST-Attention Block in GMAN, and project $STE_P \in \mathbb{R}^{P \times N \times D}$ as key, $STE_Q \in \mathbb{R}^{Q \times N \times D}$ as query, and the concatenation of X_{ADB} and STE_P as value. STE_P and STE_Q are decomposed from X_{STE} based on the historical P and future Q time steps, respectively. Then the final output $X_{STAB} \in \mathbb{R}^{Q \times N \times D}$ of the STAB is obtained.

It is worth mentioning that the proposed ADSTA-Net uses Wing loss [22] as the loss function, which helps to capture the small-size errors between the ground truth and predicted result, making the model training smoother. The specific definition is shown in *Equation 13*:

$$wing(x) = \begin{cases} \omega \, ln(1 + |x|/\varepsilon), & if |x| < \omega \\ |x| - C, & otherwise \end{cases} \tag{13}$$

where x is the error between the ground truth and predicted result, ω limits the range of the nonlinear part, ε limits the curvature of the nonlinear region and $C = \omega - \omega \, ln(1 + \omega/\varepsilon)$ is a constant.

3. EXPERIMENTS

3.1 Datasets

We conduct the experiments on two real-world datasets, which are summarised in *Table 2*: (1) XMBRT is a passenger flow dataset collected from the Automatic Fare Collection (AFC) in Xiamen BRT, with 44 stations covering 92 days from 1 March 2019 to 31 May 2019, including weekdays and weekends. Its time interval is 5 minutes, and the daily service hours are from 6:30 to 22:10, which generates 12×15+8=188 samples per day. Therefore, there are a total of 188×92=17,296 samples. (2) BJMetro is a passenger flow dataset collected from the Beijing subway, with 276 stations covering five consecutive weeks from 29 February 2016 to 3 April 2016, only including weekdays. Its time interval is 10 minutes, and the daily service hours are from 5:00 to 23:00, which generates 6×18=108 samples per day. Therefore, there are a total of 108×25=2,700 samples. On both datasets, we apply Z-Score Standardization to the data to accelerate the convergence of the model.

Table 2 – Dataset description

Description	XMBRT	BJMetro
Stations	44	276
Time span	1 March 2019 – 31 May 2019	29 February 2016 – 3 April 2016
Including weekends	Yes	No
Days	92	25
Time interval	Passenger flow: 5 minutes, Weather: 1 hour	Passenger flow: 10 minutes, Weather: half an hour, Air quality: 1 hour
Samples	17,296	2,700
External factors	11 weather indicators	4 weather indicators and 7 air quality indicators

In terms of external factors, we use 11 weather indicators on XMBRT provided by China Meteorological Administration [24], including pressure (hPa), temperature (℃), relative humidity (%), rainfall in the past hour (mm), average wind direction in 2 minutes (degree), average wind speed in 2 minutes (m/s), wind direction in maximum wind speed (degree), maximum wind speed (m/s), wind direction in extreme wind speed (degree), extreme wind speed (m/s) and ground temperature (℃), with the time interval of one hour. To maintain consistency between passenger flow and weather data for effective fusion, we reused the 12 identical hourly weather data and pre-process them using Min-Max Normalization.

We directly use the preprocessed public dataset of BJMetro [4], which already include the aligned passenger flow data and external factor data. Herein, we only use the inflow with a 10-minute interval, four weather indicators with a half-hour interval, including temperature (℃), dew point temperature (℃), relative humidity (%) and wind speed (m/s), as well as seven air quality indicators with a 1-hour interval, including real-time air quality index, PM2.5, PM10, SO_2, NO_2, CO and O_3.

3.2 Baselines

In this paper, we compare ADSTA-Net with four types of models: (1) graph convolution network (GCN)-based models, including Spatio-Temporal Synchronous Graph Convolutional Networks (STSGCN) [25], (2) Recurrent Neural Network (RNN) and GCN (RNN+GCN)-based models, including Temporal Graph Convolutional Network (TGCN) [26], Adaptive Graph Convolutional Recurrent Network (AGCRN) [27] and Meta-Graph Convolutional Recurrent Network (MegaCRN) [28], (3) attention (ATT)-based models, including Attention Based Spatial-Temporal Graph Convolutional Network (ASTGCN) [29], Spatio-Temporal Transformer Networks (STTNs) [30], Attention Based Spatial-Temporal Graph Neural Network (ASTGNN) [31], Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN) [32], GMAN [21] and Propagation Delay-Aware Dynamic Long-Range Transformer (PDFormer) [33], (4) the model considering external factors (Ex), including a deep learning architecture combining the Residual Network, GCN and Long Short-Term Memory (ResLSTM) [4].

3.3 Evaluation metric

We use mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) for evaluation, and their definitions are shown in *Equations 14–16:*

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |\hat{y}_t - y_t| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (\hat{y}_t - y_t)^2} \tag{15}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} \left|\frac{\hat{y}_t - y_t}{y_t}\right| \times 100\% \tag{16}$$

where n is the number of testing samples. $\hat{y}_t$ is the prediction result and y_t is the ground truth.

3.4 Hyperparameter settings

Table 3 lists the hyperparameter settings in ADSTA-Net. We split the datasets into training, validating and testing sets with a ratio of 7:1:2. On XMBRT, we use historical one hour passenger flow, corresponding to P=12 time steps, to predict the flow of future 15, 30 and 60 minutes, corresponding to Q=3, 6 and 12 time steps, respectively. The number of stations is N=44, and the number of external factors is E=11.

On BJMetro, we use historical one hour passenger flow, corresponding to P=6 time steps, to predict the flow of future 10, 30 and 60 minutes, corresponding to Q=1, 3 and 6 time steps, respectively. Compared with XMBRT, its number of stations is N=276, which is more complex. The number of external factors is E=11.

In the DFB, the extended dimension of $X_{Fusion}{}'$ is D=64 on both datasets. In the ADB, the extended dimension of $X_{Fusion}{}'''$ is M=256 on both datasets, and the value of $K=\lceil P/2 \rceil$ in *Equation 7* is followed by Zhou et al. [16], which is set to 6 on XMBRT and 3 on BJMetro, respectively. In the STE, L is the total number of daily time steps, which is 12×15+8=188 and 6×18=108 on XMBRT and BJMetro, respectively. In the STAB, the settings of hyperparameters are referenced from the ST-Attention Block in GMAN [21]. In Wing loss, we set ω to 5 and ε to 3, respectively.

All experiments are conducted on a server running Windows Server 2022, which is equipped with a CPU of Intel i9-13900K and a GPU of NVIDIA RTX 4090. We use the Adam optimiser with an initial learning rate of 0.001, batch size of 64 and dropout of 0.1. On the XMBRT dataset, the number of epochs is set to 200. If there is no performance improvement within 50 consecutive epochs, the optimizer will stop the training early. On the BJMetro dataset, the number of epochs is set to 1000. If there is no performance improvement within 100 consecutive epochs, the optimizer will stop the training early. In addition, all baselines except for ResLSTM [4] use solely passenger flow as their inputs. The experimental results shown in *Table 4* are obtained by running their open-source codes. Furthermore, in ResLSTM [4], P and Q on BJMetro are set to 5 and 1, respectively. For a fair comparison, we reproduce ResLSTM in our paper and adjust P to 6 and Q to 1, 3 and 6, ensuring consistency with the inputs and predicted results of all models.

Table 3 – Hyperparameter settings

Block	XMBRT	BJMetro
Dataset	P=12	P=6
	Q=3, 6, 12	Q=1, 3, 6
	N=44	N=276
	E=11	E=11
DFB	D=64	D=64
ADB	M=256	M=256
	K=6	K=3
STE	L=188	L=108
Wing loss	ω=5, ε=3	ω=5, ε=3

3.5 Main results

The main results on XMBRT and BJMetro are summarised in *Table 4.* Across all time step predictions on XMBRT, ADSTA-Net consistently achieves a slightly lower MAPE than GMAN. For the 30-minute prediction on BJMetro, ADSTA-Net achieves a slightly lower MAE than MegaCRN, and for the 60-minute prediction on BJMetro it achieves a slightly lower MAE than PDFormer. In other cases, ADSTA-Net performs better.

On the XMBRT, MegaCRN outperforms other baselines for 15-minute and 30-minute predictions, while PDFormer performs achieves the best performance for 60-minute prediction. This is due to the fact that MegaCRN employs a meta learning bank to learn traffic patterns of different types of nodes and PDFormer takes into account the time delay on propagation, thus improving prediction performance. STSGCN performs poorly, possibly due to its weak ability to simultaneously learn spatio-temporal dependencies using only GCN. Among the RNN+GCN models, AGCRN featuring an adaptive graph convolution network outperforms TGCN, which relies on stacked GRUs to expand the receptive field. This indicates that using an adaptive adjacency matrix can more effectively capture dynamic spatio-temporal features. In the ATT model, STTNs, ASTGNN and DSTAGNN achieve competitive results on most metrics. This is mainly contributed to the combination of the advanced attention mechanisms and GCN, which enables them to better extract spatio-temporal dependencies. The ASTGCN only employs the traditional attention mechanism to extract features, which limits its ability. The GMAN shows poor performance due to its reliance on using a stack of simple attention mechanisms for spatio-temporal modelling. The ResLSTM uses the second structure shown in *Figure 2b* to fuse external factors, but due to its inability to enhance the feature expression of passenger flow data, its prediction results are unsatisfactory. This strongly proves that the fusion of internal and external factors in the initial stage helps improve the model's performance.

On the BJMetro dataset, MegaCRN and PDFormer continue to outperform other baselines, while STTNs and ASTGNN achieve competitive results across most metrics. Compared with XMBRT, GMAN shows improved results on the BJMetro datatset. This could be attributed to the fact that, with more stations, the attention mechanism can better capture global spatio-temporal features.

In contrast, ADSTA-Net employs a single layer of parallel attention mechanisms to extract global spatio-temporal features, with better results than the models that combine both the attention mechanism and GCN.

This strongly proves that adaptive denoising of fused features, integrating both internal and external factors via ALPM and FFT can enhance the feature representation of passenger flow data, thus improving prediction performance.

Table 4 – Main results on the XMBRT and BJMetro dataset, Marked: 1st place, 2nd place

Dataset	**Model**		**15-min**			**30-min**			**60-min**		
			MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
XMBRT	GCN	STSGCN	6.54	10.59	44.04%	6.97	11.48	45.28%	7.86	13.40	49.15%
	RNN+GCN	TGCN	6.69	10.05	41.56%	6.68	10.11	41.61%	7.23	11.07	63.61%
		AGCRN	5.82	9.23	37.75%	5.88	9.33	37.97%	5.97	9.51	38.31%
		MegaCRN	5.44	8.39	35.78%	5.49	8.52	35.98%	5.61	8.79	36.50%
	ATT	ASTGCN	6.36	10.17	38.73%	6.52	10.46	39.82%	6.93	11.25	42.22%
		STTNs	5.70	8.97	35.98%	5.94	9.53	36.43%	6.62	11.07	37.54%
		ASTGNN	5.57	8.78	36.63%	5.66	8.98	37.07%	5.84	9.39	37.72%
		DSTAGNN	5.67	8.86	36.46%	5.66	8.88	35.95%	5.82	9.21	37.29%
		GMAN	6.08	9.46	**30.79%**	6.41	10.03	**30.53%**	7.19	11.80	**32.40%**
		PDFormer	5.53	8.53	35.40%	5.53	8.54	35.63%	5.55	8.56	35.98%
	Ex	ResLSTM	6.42	9.90	36.36%	6.20	9.48	38.67%	6.48	9.89	40.29%
		ADSTA-Net (Ours)	**5.29**	**8.13**	34.03%	**5.32**	**8.19**	35.82%	**5.34**	**8.20**	34.10%
Dataset	**Model**		**10-min**			**30-min**			**60-min**		
			MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
BJMetro	GCN	STSGCN	30.40	49.68	87.39%	32.24	55.82	87.01%	36.32	69.50	90.55%
	RNN+GCN	TGCN	72.91	138.26	143.30%	87.67	161.78	224.67%	108.35	193.81	333.06%
		AGCRN	27.05	90.09	20.40%	26.28	84.93	20.76%	27.05	86.66	22.18%
		MegaCRN	16.33	28.59	19.75%	**16.51**	28.93	20.33%	17.12	30.66	23.87%
	ATT	ASTGCN	29.74	80.16	32.26%	31.74	82.84	34.04%	36.10	92.37	40.20%
		STTNs	17.12	30.18	19.95%	17.88	32.21	21.17%	19.05	35.01	23.97%
		ASTGNN	18.49	32.51	24.70%	19.28	34.16	25.45%	20.83	37.79	30.59%
		DSTAGNN	21.45	49.84	21.10%	20.77	45.37	21.23%	21.24	45.56	22.88%
		GMAN	17.09	29.07	20.71%	23.49	43.61	23.79%	34.14	65.90	26.39%
		PDFormer	16.75	29.66	19.16%	16.81	29.74	19.13%	**17.10**	30.28	19.38%
	Ex	ResLSTM	19.39	32.45	23.22%	18.75	31.11	22.83%	20.39	34.68	23.82%
		ADSTA-Net (Ours)	**16.04**	**27.62**	**17.06%**	16.69	**28.67**	**17.61%**	17.12	**29.39**	**18.64%**

3.6 Ablation study

To verify the effectiveness and efficiency of each component in ADSTA-Net, we generate a series of variants and conduct a computational cost analysis for the tasks of future one hour passenger flow prediction on XMBRT and BJMetro. The seven variants of ADSTA-Net are summarised in *Table 5*. (1) w/o *Ex*: External factors have been removed; (2) w/o ALPM: Adaptive learning parameter matrix has been removed; (3) w/o FFT: FFT and IFFT have been removed; (4) w/o ALPM+FFT: Adaptive learning parameter matrix, FFT and IFFT have been simultaneously removed. (5) w/o *Ex*+ALPM+FFT: External factors, adaptive learning parameter matrix, FFT and IFFT have been simultaneously removed; (6) ADSTA-G: Replacing STAB with the architecture of GMAN and using the batch size of 32 on BJMetro; (7) ADSTA-L: Replacing FFT and IFFT with two linear layers in the ADB.

Table 5 – Ablation study on two datasets, marked: 1st **place**

Variant	XMBRT					BJMetro				
	MAE	RMSE	MAPE	Training time (s/epoch)	Inference time (s)	MAE	RMSE	MAPE	Training time (s/epoch)	Inference time (s)
w/o *Ex*	5.46	8.47	37.21%	/	/	17.43	30.84	19.26%	/	/
w/o ALPM	5.39	8.32	35.08%	14.7	0.4	17.49	30.62	20.48%	13.7	0.3
w/o FFT	5.42	8.38	35.09%	10.8	0.3	17.69	31.06	19.79%	8.4	0.2
w/o ALPM+FFT	5.45	8.40	39.54%	8.4	0.3	18.04	31.92	19.82%	5.7	0.2
w/o *Ex*+ALPM+FFT	5.53	8.55	36.83%	/	/	18.17	32.76	21.76%	/	/
ADSTA-G	5.87	9.13	38.07%	33.7	0.6	17.3	30.22	**17.40%**	32.7	0.7
ADSTA-L	5.38	8.28	35.00%	12.0	0.4	17.61	31.81	20.24%	9.5	0.3
ADSTA-Net (Ours)	**5.34**	**8.20**	**34.10%**	16.1	0.4	**17.12**	**29.39**	18.64%	15.6	0.4

Based on the results in *Table 5*, we can draw the following conclusions.

1) On XMBRT, observing the results of ‘w/o *Ex*’, ‘w/o ALPM’ and ‘w/o FFT’, we find that ‘w/o Ex’ yields the worst performance. This confirms our hypothesis that adding the weather indicators is indeed capable of improving the prediction accuracy. The ‘w/o FFT’ variant exhibits the second-lowest performance. It indicates that FFT can effectively reduce the noises for the fused features, which also has a certain impact on improving the performance. Although ‘w/o ALPM’ contributes the least to ADSTA-Net, ALPM can adaptively capture important features at different times and locations, which can improve the performance of ADSTA-Net to a certain extent.
2) On BJMetro, the results of ‘w/o *Ex*’ are not as poor as those observed on XMBRT. This may be because the data sample size of BJMetro is relatively smaller, so the impact of incorporating external factors is not as pronounced as that on XMBRT. Additionally, the MAE and RMSE of ‘w/o FFT’ are worse than those of ‘w/o *Ex*’ and ‘w/o ALPM’. One possible reason is that the traffic network of BJMetro is more complex, making the role of FFT more pronounced. This fully demonstrates the effectiveness of applying FFT for denoising the fused features.
3) On both datasets, we conduct the variants by removing multiple components. Except for the MAPE on XMBRT, ‘w/o Ex+ALPM+FFT’ not only performs worse than the three variants with a single component removed, but also falls behind the variant ‘w/o ALPM+FFT’ with two components removed. These results once again strongly confirm that incorporating external factors can significantly enhance ADSTA-Net’s performance. The more the components are removed, the poorer results become.
4) To more comprehensively explore the effectiveness and efficiency of ADSTA-Net, we conduct ablation study on two variants of ADSTA-G and ADSTA-L. Except for the MAPE on BJMetro, the results of ADSTA-G are not as good as those of ADSTA-Net on both datasets, which proves the effectiveness of simplifying the GMAN’s architecture. Additionally, the MAE and RMSE performance of ADSTA-G on BJMetro slightly decreases, possibly due to its more complex transportation network, where adding appropriate attention layers has a relatively small impact on model’s performance. On XMBRT, the decrease in ADSTA-L’s results is not significant, potentially due to the simplicity of the dataset with fewer stations, resulting in almost no difference in the denoising effect of using FFT or linear layers for fused features. From the perspective of training and inference time, ADSTA-G runs much slower compared to other variants and ADSTA-Net, proving that

simplifying GMAN can effectively improve computational efficiency. When conducting experiments on BJMetro with more stations in ADSTA-G, the batch size needs to be changed from 64 to 32, which strongly proves that using a multi-layer attention mechanism will significantly increase the complexity of the model. Therefore, it is necessary to simplify GMAN's architecture.

3.7 Performance and hyperparameter sensitivity analysis in Wing loss

We validate the effectiveness of Wing loss in ADSTA-Net by analysing the impact of Wing loss and three other commonly used loss functions on ADSTA-Net. All the results are future one hour prediction results on XMBRT, which is shown in *Table 6.* Evidently, Wing loss achieves the best performance across all metrics, thanks to its greater focus on small-size errors, which in turn makes the training process smoother. Consequently, we select Wing loss as the loss function used in ADSTA-Net.

In addition, Wing loss has two important hyperparameters: ε and ω. In *Figure 7*, ε is firstly set to 2 to determine the best value of ω. *Figures 7a–7c* are the results of the MAE, RMSE and MAPE, respectively The best results for MAE and RMSE are achieves when ω=2 or 5, while the MAPE performs better when ω=5 compared to ω=2. Therefore, ω is set to 5. Similarly, based on the results in *Figures 7d–7f*, we set ε to 3.

Table 6 – Performance comparison of the four loss functions on XMBRT, Marked: ***1st place***

Loss function	MAE	RMSE	MAPE	Loss function	MAE	RMSE	MAPE
Huber loss	5.35	8.23	34.26%	MAE loss	5.35	8.24	37.05%
MSE loss	5.40	8.29	38.39%	Wing loss	**5.34**	**8.20**	**34.10%**

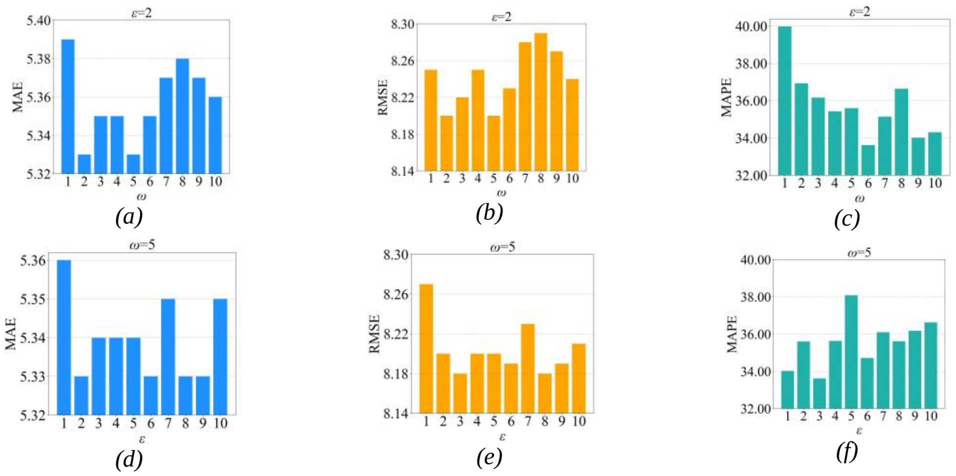


Figure 7 – Hyperparameter sensitivity analysis in Wing loss: (a) The impact of ω on MAE when ε=2; (b) The impact of ω on RMSE when ε=2; (c) The impact of ω on MAPE when ε=2; (d) The impact of ε on MAE when ω=5; (e) The impact of ε on RMSE when ω=5; (f) The impact of ε on MAPE when ω=5

3.8 Generalization analysis of the adaptive denoising

In this section, we incorporate the 11 weather indicators from XMBRT and the ADB of ADSTA-Net into the baselines STSGCN and ASTGCN to examine the impact of adding external factors on their prediction performance. Additionally, we aim to further verify the generalization ability of the adaptive denoising through ALPM and FFT within the ADB of ADSTA-Net. *Table 7* shows the comparisons of the results for the next one hour passenger flow prediction on XMBRT in the two baselines. Here, 'w/o Weather' refers to the variant without any weather indicators, while 'w/ Weather' represents the variant that includes the 11 weather indicators. We use the ratio of the absolute difference between 'w/ Weather' and 'w/o Weather' divided by 'w/o Weather' to represent the performance improvement of the two baselines after adding weather indicators and the ADB.

As shown in *Table 7*, in terms of the horizontal comparison of the results, the performance of both STSGCN and ASTGCN has been improved to a certain extent. In terms of the vertical comparison of the results, the poorer the performance of the two baselines, the more significant the improvement. Furthermore, these results also verify the generalization of the adaptive denoising.

Table 7 – Performance improvement by adding weather into the two baselines on XMBRT

Baseline	w/o Weather			w/ Weather			Improved		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
STSGCN	7.86	13.40	49.15%	6.75	11.00	42.12%	14.12%	17.91%	14.30%
ASTGCN	6.93	11.25	42.22%	6.26	10.01	40.44%	9.66%	11.02%	4.21%

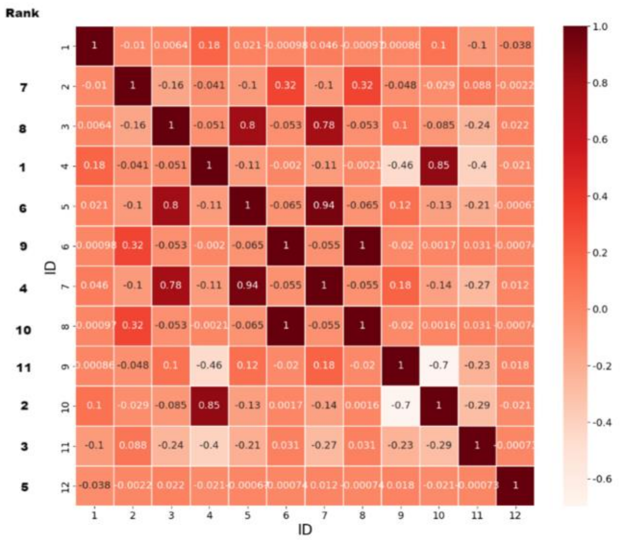
3.9 Case study

To further validate ADSTA-Net under different weather conditions, we conduct a case study on XMBRT for the future one hour passenger flow prediction. Firstly, we calculate the Pearson Correlation Coefficient between the 11 weather indicators and passenger flow to identify which indicators are more significant in influencing passenger flow on XMBRT. We then choose them to classify the weather conditions. As shown in *Figure 8*, more than half of the values are less than 0.04, indicating that most weather indicators exhibit weak correlations with passenger flow. Clearly, it is crucial to effectively denoise the fused features.

In *Figure 8*, the numbers marked in the leftmost column represent the ranking of the correlation values between each weather indicator and passenger flow. These correlations are based only on absolute values, without considering whether they are positive or negative values. Obviously, the four weather indicators most closely correlated with passenger flow are ground temperature, temperature, relative humidity and extreme wind speed.

Firstly, due to the physical similarity between ground temperature and temperature, and the fact that temperature is a more commonly used indicator in our daily life, temperature is selected. Secondly, the correlation coefficient of relative humidity ranks third and is closely related to rainfall in the past hour. Therefore, both rainfall in the past hour and relative humidity are selected. Finally, the fourth-ranked indicator, extreme wind speed, which differs from the previous three types of indicators, is also selected.

In summary, the final selected weather indicators are temperature, rainfall in the past hour, relative humidity and extreme wind speed.



ID	Factor
1	Traffic flow
2	Average wind direction in 2 minutes
3	Average wind speed in 2 minutes
4	Ground temperature
5	Maximum wind speed
6	Wind direction in maximum wind speed
7	Extreme wind speed
8	Wind direction in extreme wind speed
9	Pressure
10	Temperature
11	Relative humidity
12	Rainfall in the past hour

Figure 8 – Pearson correlation coefficient between the passenger flow and 11 weather indicators

As shown in *Table 8*, according to the four selected weather indicators, we categorise the weather conditions for weekdays and weekends into two main categories: Case 1 represents bad weather conditions, whereas Case 2 represents good weather conditions. Compared to weekends, the impact of weather conditions on travel is relatively smaller on weekdays, as weekdays are primarily dedicated to commuting for most people. Therefore, there are slight differences in the classification standards for weather conditions between weekdays and weekends, with weekdays applying stricter standards compared to weekends. Specifically, for weekdays in Case 1, the thresholds for the four weather indicators are as follows: (1) Temperature is 30°C or above. This threshold is determined by increasing the average temperature value in the testing dataset by 5°C; (2) Relative humidity is 99% or higher. This threshold is directly obtained from the testing dataset; (3) Rainfall in the past hour is 9mm/h or more; (4) Extreme wind speed is 11m/s or higher. These two thresholds are respectively determined according to rainfall and wind speed classification standards established by the China Meteorological Administration [24]. For weekends in Case 1, we only select rainfall in the past hour as the core classification criterion for weather conditions. When rainfall reaches 2mm/h or above, it is regarded as a bad weather condition. In Case 2, the weather indicator thresholds for both weekdays and weekends are reversed compared to those in Case 1.

Table 8 – Weather condition classification on XMBRT

Case \ Indicator		Temperature	Relative humidity	Rainfall in the past hour	Extreme wind speed
Case 1	Weekdays	≥30°C	≥9mm/h	≥99%	≥11m/s
	Weekends	\	≥2mm/h	\	\
Case 2	Weekdays	<30°C	<9mm/h	<99%	<11m/s
	Weekends	\	<2mm/h	\	\

To simplify the analysis, we integrate the testing data of weekdays and weekends under two different weather conditions. Namely, Case 1 includes all weekdays and weekends data under bad weather conditions and Case 2 includes all weekdays and weekends data under good weather conditions. The comparison of the predicted passenger flow for the next hour on XMBRT is shown in *Table 9*. The MAE and RMSE of Case 1 are not only better than those of Case 2, but also exceed the results of ADSTA-Net. This indicates that ADSTA-Net can predict passenger flow well even under a bad weather condition, effectively proving the robustness of ADSTA-Net. Excitingly, even when employing 11 different weather indicators without any subjective filtering, ADSTA-Net still delivers satisfactory prediction performance. The MAPEs of Case 1 is lower than those of Case 2 and ADSTA-Net. The possible reason is that the passenger flow in a bad weather condition may be smaller compared to good weather conditions. According to *Equation 16*, smaller true values can lead to larger MAPE. In addition, we remove the weather data from ADSTA-Net for a horizontal comparison. It is evident that in both cases, the results of 'w/o Weather' are worse than those of 'w/ Weather', further confirming that adding weather data significantly improves prediction performance. It also quantitatively demonstrates that changes in passenger flow are closely links to weather, thereby confirming that effectively incorporating external factors can improve the model's performance.

Table 9 – Comparison of prediction performance under the two weather conditions on XMBRT

Case	w/ Weather			w/o Weather		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
Case 1	**5.08**	**7.93**	40.32%	**5.37**	**8.45**	41.66%
Case 2	5.38	8.30	34.70%	5.50	8.53	**35.69%**
ADSTA-Net (Ours)	5.34	8.20	**34.10%**	5.46	8.46	36.09%

4. CONCLUSION

In this paper, we propose an adaptive denoising spatio-temporal attention network fused with external factors tailored for passenger flow prediction. Initially, we design a spatio-temporal information distillation module for adaptively denoising the fused features by integrating the multiple external factors with passenger flow in the initial stage of ADSTA-Net. Then, we use ALPM and FFT to achieve adaptive denoising for the fused features at different times and locations. After that, we simplify the GMAN by using only single layer ST-Attention block to further extract the global spatio-temporal features. In this way, we can simplify the complexity of the attention mechanism while guaranteeing prediction performance and reducing computational cost. Wing loss is handpicked as the loss function with the aim to better train ADSTA-Net. Extensive experiments are conducted on two real-world passenger flow datasets and the results confirm that ADSTA-Net has superior performance, particularly in making accurate predictions under bad weather conditions.

In the future, we can explore applying ADSTA-Net to other practical applications, such as optimising vehicle scheduling based on the impact of weather on passenger flow. Furthermore, we can collect additional external factor data, such as POIs and weather data corresponding to larger transportation networks, to validate the feasibility of the proposed ADSTA-Net. Additionally, we can explore more efficient methods for multi-source data fusing and denoising.

ACKNOWLEDGEMENTS

This research is partly supported by the National Natural Science Foundation of China (No. 62103345), and partly supported by Key Laboratory of Fujian Universities for Virtual Reality and 3D Visualization.

REFERENCES

[1] Cao XD, Shi Q, Chen YK, Chen CC. Short-term traffic flow uncertainty prediction based on novel GM(1,1). *Promet-Traffic&Transportation.* 2024;36(3):450–462. DOI: 10.7307/ptt.v36i3.497.

[2] Li LC, et al. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowledge-Based Systems.* 2019;172:1–14. DOI: 10.1016/j.knosys.2019.01.015.

[3] Li LC, et al. A hybrid method coupling empirical mode decomposition and a long short-term memory network to predict missing measured signal data of SHM systems. *Structural Health Monitoring-an International Journal.* 2020;20(4): 1778–1793. DOI: 10.1177/1475921720932813.

[4] Zhang JL, et al. Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems.* 2021;22(11):7004–7014. DOI: 10.1109/TITS.2020.3000761.

[5] Fang S, et al. MS-Net: Multi-source spatio-temporal network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems.* 2022;23(7):7142–7155. DOI: 10.1016/j.neunet.2023.106093.

[6] Qin XA. Traffic flow prediction based on two-channel multi-modal fusion of MCB and attention. *IEEE Access.* 2023;11:58745–58753. DOI: 10.1109/ACCESS.2023.3280068.

[7] Yu X, Bao YX, Shi Q. Dynamic multiple-graph spatial-temporal synchronous aggregation framework for traffic prediction in intelligent transportation systems. *PeerJ Computer Science*. 2024;10:e1913. DOI: 10.7717/peerj-cs.1913.

[8] Hou Y, Deng ZY, Cui HK. Short-term traffic flow prediction with weather conditions: Based on deep learning algorithms and data fusion. *Complexity*. 2021;2021:6662959. DOI: 10.1155/2021/6662959.

[9] Zhang JB, Zheng Y, Qi DK. Deep spatio-temporal residual networks for citywide crowd flows prediction. *Proceedings of the AAAI Conference on Artificial Intelligence 2017, 4-9 Feb. 2017, San Francisco, California, USA.* 2017. p. 1655–1661. DOI: 10.1609/aaai.v31i1.10735.

[10] Liang YX, et al. Fine-grained urban flow prediction. *Proceedings of the Web Conference 2021, 19-23 Apr. 2021, Ljubljana, Slovenia.* 2021. p. 1833–1845. DOI: 10.1145/3442381.3449792.

[11] Sun JK, et al. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering.* 2022;34(5):2348–2359. DOI: 10.1109/TKDE.2020.3008774.

[12] Li LC, et al. Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network. *Applied Soft Computing.* 2018;73:921–932. DOI: 10.1016/j.asoc.2018.09.023.

[13] Lin CM, Du BW, Sun LL, Li LC. Hierarchical context representation and self-adaptive thresholding for multivariate anomaly detection. *IEEE Transactions on Knowledge and Data Engineering.* 2024;36(7):3139–3150. DOI: 10.1109/TKDE.2024.3360640.

[14] Chen XY, Sun LJ. Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2022;44(9):4659–4673. DOI: 10.1109/TPAMI.2021.3066551.

[15] Kumar A, et al. Stationary wavelet transform based ECG signal denoising method. *ISA Transactions.* 2021;114(24):251–262. DOI: 10.1016/j.isatra.2020.12.029.

[16] Zhou T, et al. FiLM: Frequency improved legendre memory model for long-term time series forecasting. *Proceedings of the 36th International Conference on Neural Information Processing Systems 2022, 28 Nov-9 Dec. 2022, New Orleans, LA USA.* 2022. p. 12677–12690.

[17] Kim SM, Kim YS. Enhancing sound-based anomaly detection using deep denoising autoencoder. *IEEE Access.* 2024;12:84323–84332. DOI: 10.1109/ACCESS.2024.3414435.

[18] Liu Z, Ma QL, Ma PT, Wang LH. Temporal-frequency co-training for time series semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence 2023, 7-14 Feb. 2023, Washington DC, USA.* 2023. p. 8923–8931. DOI: 10.1609/aaai.v37i7.26072.

[19] Ilias L, et al. Unsupervised domain adaptation methods for photovoltaic power forecasting. *Applied Soft Computing.* 2023;149:110979. DOI: 10.1016/j.asoc.2023.110979.

[20] Guo SN, et al. Deep spatio-temporal 3D convolutional neural networks for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems.* 2019;20(10):3913–3926. DOI: 10.1109/TITS.2019.2906365.

[21] Zheng CP, Fan XL, Wang C, Qi JZ. GMAN: A graph multi-attention network for traffic prediction. *Proceedings of the AAAI Conference on Artificial Intelligence 2020, 7-12 Feb. 2020, Hilton Midtown, New York, USA.* 2020. p. 1234–1241. DOI: 10.1609/aaai.v34i01.5477.

[22] Feng ZH, et al. Wing loss for robust facial landmark localization with convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition 2018, 18-23 Jun. 2018, Salt Lake City, UT, USA.* 2018. p. 2235–2245. DOI: 10.1109/CVPR.2018.00238.

[23] Xu JJ, et al. Understanding and improving layer normalization. *Proceedings of the 33rd International Conference on Neural Information Processing Systems 2019, 8-14 Dec. 2019, Vancouver, Canada.* 2019. p. 4381–4391.

[24] China Meteorological Administration, 2024. https://www.cma.gov.cn/en/.

[25] Song C, Lin YF, Guo SN, Wan HY. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence 2020, 7-12 Feb. 2020, Hilton Midtown, New York, USA.* 2020. p. 914–921. DOI: 10.1609/aaai.v34i01.5438.

[26] Zhao L, et al. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems.* 2020;21(9):3848–3858. DOI: 10.1109/TITS.2019.2935152.

[27] Bai L, et al. Adaptive graph convolutional recurrent network for traffic forecasting. *Proceedings of the 34th International Conference on Neural Information Processing Systems 2020, 6-12 Dec. 2020, Vancouver, Canada.* 2020. p. 17804–17815.

[28] Jiang RH, et al. Spatio-temporal meta-graph learning for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence 2023, 7-14 Feb. 2023, Washington DC, USA.* 2023. p. 8078–8086. DOI: 10.1609/aaai.v37i7.25976.

[29] Guo SN, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence 2019, 27 Jan-1 Feb. 2019, Honolulu Hawaii, USA.* 2019. p. 922–929. DOI: 10.1609/aaai.v33i01.3301922.

[30] Xu MX, et al. Spatial-temporal Transformer networks for traffic flow forecasting. *Arxiv Preprint,* 2020. DOI: 10.48550/arXiv.2001.02908.

[31] Guo SN, et al. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering.* 2021;34(11):5415–5428. DOI: 10.1109/TKDE.2021.3056502.

[32] Lan SY, et al. DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. *Proceedings of the 39th International Conference on Machine Learning 2022, 17-23 Jul. 2022, Baltimore, Maryland USA.* 2022. p. 11906–11917.

[33] Jiang JW, Han CK, Zhao WX, Wang JY. PDFormer: Propagation delay-aware dynamic long-range Transformer for traffic flow prediction. *Proceedings of the AAAI Conference on Artificial Intelligence 2023, 7-14 Feb. 2023, Washington DC, USA.* 2023. p. 4365–4373. DOI: 10.1609/aaai.v37i4.25556.