STASG: a Novel Traffic Prediction Model Based on Spatial-Temporal Attention Simple Graph Neural Network

Xiujuan Xu[1], Jiaxin Ai[1], Renjie Liu[1], Xiaowei Zhao[1](✉) and Yu Liu[1]

[1] School of Software Technology, Dalian University of Technology, Dalian, China

{xjxu, xiaowei.zhao, yuliu}@dlut.edu.cn

Abstract. The growth of the autonomous driving industry in recent years has spurred research on intelligent transportation systems. However, predicting long-term traffic patterns is a complex task that can lead to overfitting and fluctuations in model predictions. To address these challenges, this paper proposes a spatio-temporal modeling approach called STASG that captures both the spatial and temporal features of traffic data. The method fuses these features using a gated fusion mechanism and then applies feedforward neural networks to transform the spatio-temporal data into predictions for future time steps. To mitigate overfitting, the paper introduces a novel loss function called the mean loss function. By minimizing fluctuations in model predictions, this approach aims to improve the accuracy of long-term traffic forecasts. Overall, this paper presents a promising approach to improving the performance of intelligent transportation systems, particularly in the area of long-term traffic prediction. The proposed method combines several techniques, including spatio-temporal modeling, neural networks, and a new loss function, to address the challenges of overfitting and prediction fluctuations. After conducting multiple experiments on the publicly available transportation network datasets, METR-LA and PEMS-Bay, our proposed model demonstrated improved performance in long-term traffic flow prediction.

Keywords: Traffic prediction · Gated attention unit · Mean Value loss · Graph neural network.

1 Introduction

With the advancement of GPS and sensor technology, the proliferation of spatiotemporal data has surged from diverse sources such as mobile phones, car navigation systems, and traffic sensors. Urban traffic forecasting, which serves as a fundamental research topic, has laid a foundation for exploring the dynamic characteristics of urban traffic networks. Traffic forecasting provides a significant technical basis for intelligent transport [1], low-carbon city construction, and urban traffic management.

The objective of traffic flow data prediction is to leverage historical time-step traffic flow and other sensor-gathered features to predict future traffic conditions. Given the rapid growth of autonomous driving [2], electronic maps [3], and other transportation industries, solving this problem has become a crucial strategy to reduce traffic congestion, optimize driving routes, enhance road efficiency, and tackle road traffic issues. Moreover, long-time traffic flow prediction is a prerequisite and fundamental basis for intelligent traffic systems [4]. In recent years, deep learning has gained extensive attention in diverse industries for predictive modeling of data.

To address spatial modeling of images and temporal modeling of time series, deep learning researchers have devised convolutional neural networks and their deformations [5] as well as recurrent neural networks and their deformations [6, 7]. Nevertheless, traffic data represents non-Euclidean structured data, which employs an adjacency matrix to depict the position relationship between sensor nodes. The spatial features of this non-Euclidean structure cannot be captured by simple convolutional neural networks. The emergence of graph neural networks [8, 9] has brought a completely new research perspective to this field. Although graph neural networks [10, 11] have been utilized to model the structure of traffic maps, the spatio-temporal nature of traffic flow data presents a significant challenge to this research.

In our quest to dissect and understand the complex world of sensor networks and traffic flow, we've masterfully combined an adaptive adjacency matrix with a traditional sensor node adjacency matrix, enabling us to not only uncover the static spatial attributes of sensor nodes but also reveal their hidden spatial connections. Tackling the challenge of over-smoothing inherent in multi-layer graph convolution, we've streamlined the process with a graph convolutional neural network adept at capturing spatial features by embracing the insights from multi-hop neighbor nodes. For temporal analysis, the Gate Attention Unit stands at the forefront, elegantly preserving global feature interaction while simplifying the complexity typically associated with attention mechanisms, thus enhancing processing speed. Our model is the use of a gating mechanism that intelligently merges spatio-temporal features, adeptly mimicking the complex interplay of space and time within traffic dynamics. This holistic approach not only sheds light on the intricacies of traffic flow but also paves the way for advanced models capable of predicting the rhythmic dance of urban traffic with unprecedented precision.

2 Related Work

2.1 Traffic Prediction

Whether it is a short-term traffic forecasting or long-term traffic forecasting problem, the focus is on a data-driven approach, i.e., forecasting based on historical data. The traffic forecasting problem is more challenging than other time series forecasting problems because it involves large data volumes with high dimensionality and data transformation in different dimensions, specifically large data volumes with high dimensionality and multiple dynamic factors, including emergency situations such as traffic accidents. Examples include historical average and integrated moving average (ARIMA) models [12]. This spatio-temporal prediction problem cannot be handled.

Machine learning (ML) and deep learning techniques have been introduced in this field to improve prediction. Gridding data in spatial domain using CNN [13], Using RNN and its deformation LSTM in time distribution prediction [14, 15]. However, all these modeling approaches have shortcomings. For CNNs, it ignores that the underlying graph structure of traffic data is non-Euclidean. At the same time, RNNs and their deformations cannot effectively exploit the spatial properties of traffic data for long-term traffic prediction. Recent research has built traffic prediction on graphs and used GNNs to model non-Euclidean structures in road networks. GNNs can capture complex relationships between objects and make inferences based on the described data as a neural network acting directly on graph structures. GNNs are effective for node-level, edge-level, and graph-level prediction tasks in various situations. GNNs are currently considered state-of-the-art for traffic prediction problems, and these non-Euclidean structure-based models generate future traffic data through a multi-step prediction approach [17, 18].

2.2 Self-Attention Mechanism

The attention mechanism was first introduced in the field of natural language processing, and the popularity of the Transformer [19] has led to a wide range of applications, including the Transformer in machine translation, the BERT [20], and GPT families of pre-training models [21, 22]. It is also used in computer vision in the Vision Transformer [23] and Swin Transformer [24]. It eliminates the problem of gradient disappearance or gradient explosion when training models with long data sequences and can effectively exploit the parallel computing power of GPUs to improve the training speed and prediction accuracy significantly. In addition, there has been some research on using attention mechanisms in traffic prediction. Using appropriate spatio-temporal location embeddings, researchers have captured the spatio-temporal characteristics of traffic data using attention mechanisms to predict future changes in traffic conditions by annotating the location information of spatio-temporal sequences. For example, GeoMAN [25] is the first to introduce a multi-layer attention mechanism to the spatio-temporal data prediction problem, modeling the dynamic spatio-temporal correlations between sensors, and GMAN [26] adds a transformed attention layer between Encoder and Decoder to transform the encoded historical features to generate future feature representations. Our capture of temporal dependencies is an improvement on this [36]. Nonetheless, since the computation of the previous temporal self-attention mechanism is too computer resource-intensive, we slightly improve the existing self-attention mechanism by structurally fusing it with gated linear units to achieve computational efficiency [37].

2.3 Graph neural network

For graph data graph neural network is a network with good learning performance, which can extract a large amount of spatial information from non-Euclidean structured data, thus providing new opportunities for transportation prediction. Based on graph theory, nodes, and edges have features that can be convolved or aggregated. These features reflect various traffic conditions, such as traffic volume, speed, number of lanes, and road class, spatial data. Graph neural networks can be seen as an extension of deep neural networks to graph data [27]. Since graph structures are not traditional grid-structured data, traditional deep neural networks cannot be extended to graph-structured data. Using node features and graph structures as input, the researchers aim to learn representative features of each node, which can be of great help in subsequent classification and regression tasks.

3 Methodology

3.1 Problem Definition

We assume that there are n sensor nodes N on the road traffic network, each sensor node can record the traffic flow characteristics of the current location in real time (in this paper the characteristics refer to the traffic flow S), the traffic condition in a certain time period can be expressed as X_t belonging to $R^{n \times s}$. The purpose of the traffic prediction problem in a time t is to obtain a model f trained by the historical time period p, which can predict the traffic conditions in the future time period q,The specific formula 1 is shown below

$$X_{t_1}, X_{t_2}, \cdots, X_{t_p} \xrightarrow{\mathrm{f}} X_{t_{p+1}}, X_{t_{p+2}}, \cdots, X_{t_{p+q}} \tag{1}$$

3.2 Framework of STASG

As shown in Fig.1, we present our proposed framework with a graph attention-based traffic prediction network model (STASG). Similar to the mainstream spatio-temporal prediction models nowadays, STASG also adopts an encoder-decoder structure, and our model consists of four parts, which are spatiotemporal embedding location encoding, graph adaptive adjacency matrix, encoder and decoder modules, and the output of the final decoder, which first undergoes a nonlinear transformation by a two-layer mean prediction network to generate the mean of traffic features for future time step prediction, is also input to a layer of The final result is obtained by the dimensional transformation of the full connection. The spatio-temporal embedding location coding can extract the features of our input traffic flow and fuse the temporal and spatial features in the original traffic data to obtain the spatio-temporal features that can be more easily recognized by the encoder and decoder. The spatio-temporal node features are obtained by summing the adaptive adjacency matrix and the known adjacency matrix during the training process and inputting them into the spatial graph convolution module to obtain more flexible spatial node features. n spatio-temporal modules and two consecutive fully connected layers are included in each of our encoders and decoders respectively, and each spatio-temporal module we adopt a gating mechanism to fuse the temporal gated attention unit and the spatial simple. The spatio-temporal features generated by the convolutional unit of the graph. The final STASG model requires the output dimension of each module to be the same as the input data dimension because it is convenient to add jump connection structure to each module to break the symmetry of the model structure, which is used to alleviate the model degradation problem common to neural network models.

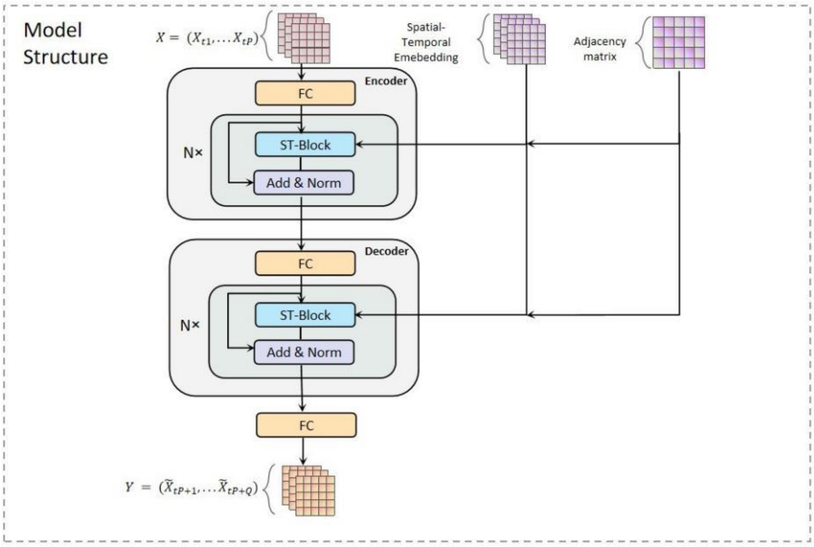

Model Structure
$X = (X_{t1}, \dots X_{tP})$
Spatial-Temporal Emebedding
Adjacency matrix
Encoder
FC
N×
ST-Block
Add & Norm
Decoder
FC
N×
ST-Block
Add & Norm
FC
$Y = (\widetilde{X}_{tP+1}, \dots \widetilde{X}_{tP+Q})$

Fig. 1. The structure of our model STASG

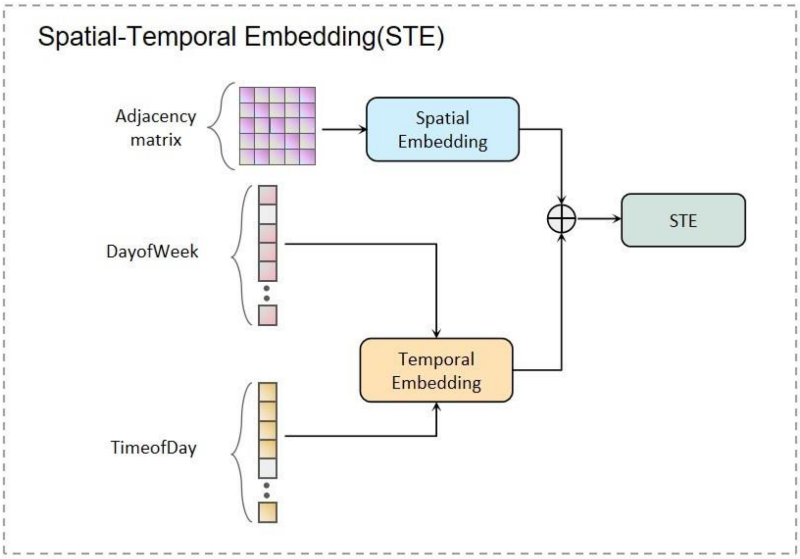


Fig. 2. Structure of spatial embedding and temporal embedding

3.3 Spatio-temporal embedding position matrix

The variation of spatio-temporal correlation of traffic prediction data mainly depends on the difference of temporal nodes and the different locations of spatial nodes. The self-attentive mechanism can be more effective than the RNN series model and its deformation in capturing the corresponding temporal features. However, in the RNN series model, the input features naturally carry sequential features, and the self-attentive mechanism obviously does not have the advantage in this regard. As shown in Fig.2, Encode the original road network data and the adjacency matrix data with spatial and temporal embedding locations. Firstly, in the spatially embedded location coding, we use the node2vec method to learn the distance and connectivity of sensor nodes with fixed point representation, and the previously trained vector of each node will be trained adaptively as the model runs, and at the end of each training session, the training enters the two-layer 2D convolutional network for dimensional transformation to obtain the spatially embedded representation vector S_e subset R^D, and in this paper we use S_e for the representation. The specific formula 2 is shown below:

$$S_e = conv_{1\times1}\left(conv_{(1\times1)}\big(node2vec(adj, distance)\big)\right) \tag{2}$$

adj represents the connectivity adjacency matrix between each sensor node, and distance represents the distance matrix between different sensor nodes.

The spatial embedding representation vector can only represent the regional connectivity representation of the whole dataset, when our traffic dataset is used as a static graph, the spatial embedding vector is sufficient. Most of the spatio-temporal characteristics of the traffic flow prediction dataset are dynamic, and based on this feature, we adopt a temporal embedding method. Firstly, we extract the temporal features of the data by using the unique thermal coding method. Each data is encoded with two temporal locations, $day_{position}$ for 7 days in a week and $point_{position}$ for 24 hours in a day. The two global location codes are concatenated to obtain the initial temporal embedding representation vector, The specific formula 3 is shown below:

$$\widehat{T_e} = concat(\mathrm{day}position, \mathrm{point}position) \tag{3}$$

Next, we apply a two-layer two-dimensional 1×1 convolutional network to transform the dimensionality of the time-embedded vector to d dimensions.

$$T_e = conv_{1\times1}\left(conv_{1\times1}\big(\widehat{T_e}\big)\right) \tag{4}$$

To obtain dynamic and static representations of traffic data, we sum the temporal embedding representation vector and the spatial embedding representation vector mentioned above to obtain the temporal embedding vector ST_e that fuses the features of both for the subsequent temporal gating attention mechanism.

$$ST_e = S_e + T_e \tag{5}$$

3.4 Graph Adaptive Adjacency Matrix

We define a self-learning adjacency matrix A_{learn} in addition to the traditional adjacency matrix of sensor nodes in order to preserve the generalization of the spatial structure data.

$$A_{learn} = softmax\big(relu(A_1 A_2)\big) \tag{6}$$

where $A_1 \in R_{N\times c}$ and $A_2 \in R_{N\times c}$ denote two randomly initialized learnable parameters. We use the $relu()$ function to maintain the nonlinear variation of the self-learning adjacency matrix. Finally, the self-learning adjacency matrix is normalized using the *softmax* function. The normalized self-learning adjacency matrix can be used as a dynamic representation of the current spatial state. Then the random initialized self-learning adjacency matrix A_{learn} is fused with the initial adjacency matrix $A_{support}$. This fusion operation can be defined as follows:

$$A_{adj} = A_{support} + A_{learn} \tag{7}$$

The fused adjacency matrix A_{adj} is input to the spatial graph convolution module to simulate the spatial features of each road. The adaptive adjacency matrix A_{adj} can dynamically complete the missing graph structure in the initial adjacency matrix to improve the accuracy of model prediction by continuously improving the graph structure.

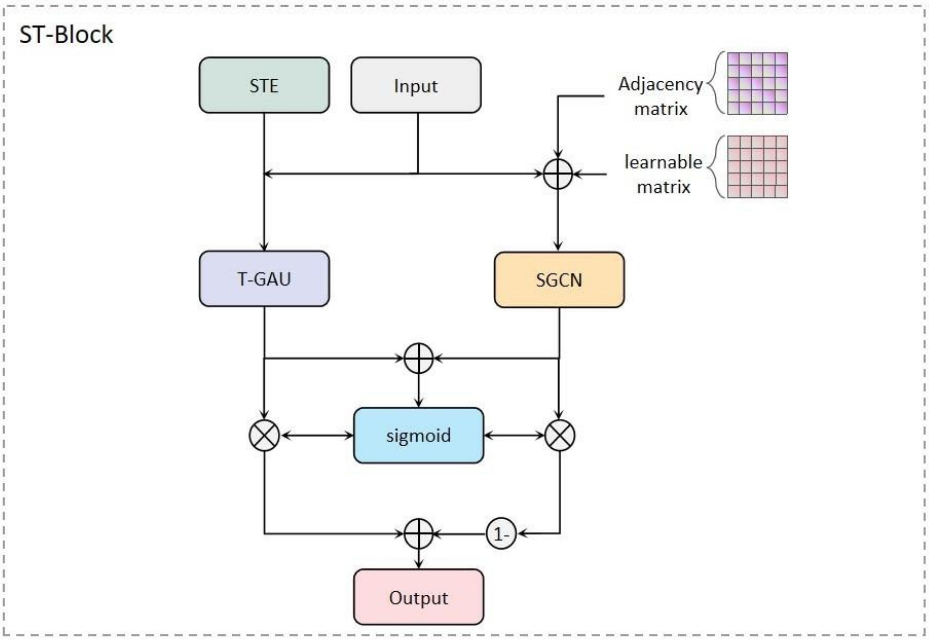


Fig. 3. Structure of ST-Block

3.5 ST-Block

As shown in Fig.3, the ST-Block we define consists of a temporal gated attention unit, a spatial graph convolution unit and a gated fusion unit. We denote the output of the m-th block as $H^{(m-1)}$, where the hidden state of vertex v_i at time step t_j is denoted as $H^{(m-1)}$. The outputs of the time-gated attention and spatial graph convolution units of the m-th block are denoted as $H_S^{(m)}$ and$H_T^{(m)}$, respectively, where the temporal hidden states of the vertex v_i at the time step t_i are denoted as $h_{v_i,t_j}^{t(m)}$ and $h_{v_i,t_j}^{s(m)}$. Finally, we input the temporal hidden state and spatial hidden state to the gated fusion unit, denoted as $H^{(m)}$.

Temporal Gated Attention Unit In the past, spatio-temporal prediction models usually use RNN and its deformations, especially Long Short Time Memory (LSTM) and Gated Recursive Unit (GRU) are applied to model temporal correlation, although the RNN model establishes the temporal correlation of each unit, we can also see that the LSTM model widely identifies the nonlinear variation of traffic data and effectively eliminates the gradient decrement, but cannot identify the temporal periodicity and dynamic trend of real-time traffic data. Therefore, we propose a temporal gated attention mechanism in ST-Block, which not only captures the temporal trend and periodicity, but also dynamically obtains global temporal information in multiple time ranges, and focuses attention on the most important information of current traffic prediction, which has a faster speed and lower memory consumption than the temporal multiheaded attention units proposed by other similar models. It has faster speed, lower memory consumption, and better results than other similar models. We find that the current time traffic flow prediction has a non-linear correlation with the previous multiple time steps. We design an adaptive modeling of the temporal characteristics of temporal data for the past *t* time steps by fusing an attention mechanism with a gated linear unit. Specifically, we connect the current state with a temporal embedding matrix to adaptively model the nonlinear correlation between different time steps and use a gated attention method to calculate the attention scores. Formally, we first consider the correlation between the vertex vi, the other nodes v using the multi-headed attention approach defined as:

$$\hat{X} = concat(X, ST_e) \tag{8}$$

$$Score_{v_i,v}^{(k)} = \frac{\left(FC_q\left(\hat{x}_{v_i,t_j}\right)\left(FC_k\left(\hat{x}_{v_i,t_j}\right)\right)\right)}{\sqrt{d}} \tag{9}$$

$$h_{v_i,v}^{(k)} = softmax\left(Score_{v_i,v}^{(k)}\right) \cdot FC_v\left(\hat{x}_{v_i,t_j}\right) \tag{10}$$

$\hat{X}$ consists of the initial data X and the spatio-temporal embedding vector through the connection function $concat()$, $FC_q()$, $FC_k()$, $FC_v()$ denote three different nonlinear mappings, $\hat{x}_{v_i,t_j}$ and $FC_k()$, mapping, multiply to get the similarity of the node, divide by $\sqrt{d}$ to get $Score_{v_i,v}^{(k)}$, use softmax function to normalize the exponential, multiply with the mapping of $FC_v()$, and get the hidden state $h_{v_i,v}^{(k)}$.

The use of multi-headed temporal attention mechanism can calculate the temporal correlation between each vertex, but it requires more time and space overhead due to the global calculation. We can analyze the structure of the gated linear unit and the multi-headed attention mechanism through the model structure, and the gated linear unit can not realize the interaction between different vertex time features.

In T-GAU $FC_q(), FC_k()$ represent two different simple affine transformations, i.e., multiplying by a trainable parameter γ, and adding a trainable parameter β. The $relu^2$ () used here is equivalent to $relu^2$, which normalizes the obtained attention matrix by using the attention fraction A_{v_i,v_j} by adaptively selecting the relevant temporal features over all historical P time steps,, and to avoid neuron death during training, in FC_u , FC_v we use $silu()$ instead of $relu()$ in the original GLU to perform the nonlinear transformation. Finally, we get the temporal features of the corresponding nodes $h_{v_i,v}^{(k)}$. The use of multi-headed temporal attention mechanism can calculate the temporal correlation between each vertex, but it requires a huge time and space overhead due to the global computation. We can analyze the structure of the gated linear unit and the multi-headed attention mechanism through the model structure, which has many similarities.

$$A_{v_i,v} = relu^2 \frac{FC_q\left(\hat{x}_{v_i,t_j}\right)FC_k\left(\hat{x}_{v_i,t_j}\right)}{\sqrt{s}} \tag{11}$$

$$h_{v_i,v,t_j} = \left(\left(FC_u\left(\hat{x}_{v_i,t_j}W_u\right) \odot \mathrm{A} \cdot FC_v\left(\hat{x}_{v_i,t_j}W_v\right)\right)\right)W_h \tag{12}$$

For temporal feature capture, we use the original data and spatio-temporal embedding position encoding to aggregate the feature information from different temporal levels after transformation. We incorporate the temporal attention mechanism into the gated linear unit for better efficiency of the model, and define the temporal attention gated unit, which not only has a substantial speedup over the temporal attention mechanism, but also obtains better time-transformed capture results.

Simple Spatial Graph Convolution Unit Arranging neurons into small convolutional kernels that traverse local spatial locations is the key to CNN success. The feasibility of this design lies in the Euclidean structure of image data, allowing matrix convolutions, and the translation invariance of image objects. Therefore, spatial translation relations are not useful at all in the spatial modeling of many graphs. The graph Fourier transform first fuses the structural data with the adjacency of the represented edges using a Laplacian matrix, and this fused information can be easily transformed into the spectral domain due to the properties of the Laplacian matrix i.e., a spectral decomposition. The nodes can be projected into the spectral domain space, thus completing the combination of the three. Then we can do the convolution operation in the spectral domain, and after the operation, we can go back to the null domain. We originally used the spatial map convolution unit to extract spatial features $h_s^{(l+1)}$ from the input initial road network data and the adaptive adjacency matrix:

$$h_s^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}h_s^{(l)}W^{(l)}\right) \tag{13}$$

$$\tilde{A} = A + I \tag{14}$$

where A represents the adjacency matrix, $h_s^{(l)}$represents the hidden state of the lth layer, $W^{(l)}$ represents the spatial weight matrix randomly initialized at the lth layer, I represents the unit matrix, D is the degree matrix of the node, and σ represents the nonlinear activation function $relu()$.

However, we found after several experiments that when we stack multiple layers of GCNs, we find that the performance of our model degrades sharply. Current spatial graph convolution models in the traffic domain are using shallow GCN stacking architectures, which we believe not only leads to limitations in the expressiveness of GCNs, but also constrains the deep mining of spatial features for traffic data. Research on graph neural networks has found that when GNN models are stacked with multiple layers, there is always a phenomenon that the output representation of nodes becomes indistinguishable, i.e., the problem of over-smoothing. Some researchers found that while expanding the number of convolution layers of large input graphs and adjacency matrices alone increases the risk of the over-smoothing problem, the performance is only slightly affected, but if the number of nonlinear transformation layers is increased simultaneously, the performance will drop sharply. We therefore decided to mitigate the model performance degradation caused by increasing the number of network layers by removing the nonlinear layers between the GCN layers and collapsing the resulting hidden states into a single nonlinear transformation:

$$h_s^{(l+1)} = \left(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}\right)^l XW \tag{15}$$

$$H_s = \sigma\left(h_{s_{end}}\right) \tag{16}$$

X represents the input road network data, $\tilde{A}$ represents the adaptive adjacency matrix, W represents the weights uniformly initialized by the simple spatial graph convolution unit, l represents the number of layers of the graph convolution, $h_{s_{\mathrm{end}}}$ represents the hidden state of the last layer, and the σ function uses sigmoid(). The simple spatial graph convolution unit increases the number of layers in capturing deep spatial features while maintaining the same number of layers in the perceptual field.

3.6 Gated Fusion Mechanism

The traffic conditions of the current road at the current time step are highly correlated with the traffic conditions of all other roads in the previous traffic. We design a gating mechanism to adaptively fuse spatial and temporal features. In the lth block, the outputs of the spatial and temporal attention mechanisms are denoted as $H_s^{(l)}$ and $H_t^{(l)}$ respectively, in the encoder as $R^{P\times N\times D}$ and in the decoder as $R^{Q\times N\times D}$. The $H_s^{(l)}$ and $H_t^{(l)}$ are fused as:

$$H^{(l)} = z \odot \ H_s^{(l)} + (1 - z) \odot \ H_t^{(l)} \tag{17}$$

$$z = sigmoid\left(H_s^{(l)} W_s + H_t^{(l)} W_t + b_z\right) \tag{18}$$

where $W_s \in R^{D \times D}$, in $W_t \in R^{D \times D}$, $b_z \in R^D$are learnable parameters, $\odot$ denotes Hadamard product, and z is the gating coefficient controlling the distribution of spatio-temporal features of the traffic flow. The gated fusion mechanism adaptively controls the spatially and temporally dependent flow of each vertex and timestep.

3.7 Mean Scope Loss Function (MSL)

We found that the model is mean shifted when predicting traffic flow in future time periods. To cope with this situation, we add a mean penalty term to the original mean absolute relative error (MAE). We first add a dimensionally transformed fully connected layer FC_m outside the STASG network architecture for predicting the mean Mean at the next n time steps. and then subtract it from the true mean to obtain the mean penalty term $\widehat{Mean}$. We set a hyperparameter γ as the weight coefficient of the penalty term. Finally, it is added with MAE to get the final mean error Mean_MAE:

$$Mean = FC_m(STASG(X)) \tag{19}$$

$$\widehat{Mean} = y\_Mean - Mean \tag{20}$$

$$Mean\_MAE = MAE + \widehat{Mean} \tag{21}$$

Where Mean represents the predicted mean of STASG, y_Mean represents the true mean of the future time step, and MAE represents is the mean of the absolute error.

4 Experimental Methodology

In this section, we give experimental results of STASG and baseline models on traffic datasets, i.e., Metra-LA, PEMS-Bay, published by Li et al. We also analyze the model performance of different types of attention and model configurations for ablation studies.

4.1 Datasets

METR-LA recorded four months of traffic speed statistics for 207 sensors on Los Angeles County freeways. PEMS-BAY contains six months of traffic speed information for 325 sensors in the Bay Area. Detailed distribution information for both datasets is shown in Fig. 6. We used the same data preprocessing procedure as other traffic prediction methods. As a time window of traffic flow, we use the sensor's recorded traffic flow every five minutes. The adjacency matrix of nodes is constructed from the road network distance with a threshold Gaussian kernel. z-score normalization is applied to the input. The dataset is divided chronologically, 70% for training, 10% for validation, and 20% for testing. The detailed dataset statistics are shown in Table 1.

Table 1: Introduction of METR-LA and PEMS-BAY datasets

Data	Nodes	Edges	Time Steps
PEMS-BAY	325	2369	52116
METR-LA	207	1515	34272

4.2 Experimental Settings

Our experiments were conducted on a GPU server with two GeForce GTX 3090Ti graphics cards. As a benchmark evaluation, the following settings were kept constant for each model. Both training and prediction steps were set to 3,6,12. The ratio of data for training, validation, and testing was set to 7:1:2. Adam was set as the default optimizer, where the learning rate was set to 0.001 and the batch size was set to 64 by default. the average absolute error was unified as a loss function. If the validation error converges within 20 calendar hours, the training algorithm is stopped early or after 200 calendar hours and the best mod on the validation data is saved. mods are trained for prediction using L=2 and L=3, and L=3 is used as the main comparison criterion. root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) are used as inference-time measures, where zero values will be ignored.

4.3 Baselines

ARIMA [12]: An autoregressive sliding average model based on Kalman filtering.
HistoricalAverage [28]: Single-step prediction model based on inflow and outflow of grid traffic data.
STGCN [29]: Based on a combined graph convolution and dimensional convolution model.
LSTNet [30]: The model based on long and short-term memory neural networks for the construction of long-term and short-term temporal patterns.
GMAN [26]: An encoder-decoder structure is used. Both the encoder and decoder are composed of multiple spatio-temporal attention modules combined with a gating mechanism to create a combination of spatio-temporal factors.
Graph WaveNet[31]: A novel adaptive dependency matrix is introduced that allows learning inference by using graph convolution on spatial node embeddings.
ASTGCN [32]: Combines a spatio-temporal attention mechanism while capturing the dynamic spatio-temporal characteristics of traffic data using convolution.
DCRNN [33]: Combining graph convolutional networks with recurrent neural networks in an encoder-decoder fashion.

MTGNN [35]: A novel graph neural network framework for multivariate time series forecasting introduces an adaptive dependency matrix enabling the automated extraction of variable relations and the incorporation of external knowledge.
AGCRN [34]: an encoder-decoder model, integrates node-adaptive and data-adaptive modules with a recurrent network to extract fine-grained spatio-temporal factors for enhanced traffic prediction Our model performance is compared with other benchmark models in the PEMS-BAY dataset. The MAPE metric of STASG in 6 time steps is slightly worse than Graph WaveNet, and the other time steps are the best among other models.

4.4 Results And Discussion

Table 2. Comparison of the performance of individual models on the PEMS-Bay dataset.

Model	Timestep: 15MIN			Timestep: 30MIN			Timestep: 60MIN		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
HistorialAverage	3.31	6.68	8.09%	3.33	6.69	8.09%	3.33	6.68	8.10%
LSTNet	1.66	3.29	3.52%	2.22	4.37	5.12%	2.76	5.17	6.10%
STGCN	1.36	2.87	2.86%	1.72	3.90	3.93%	2.04	4.69	4.87%
GMAN	1.28	2.71	2.71%	1.47	3.21	3.23%	1.87	4.31	4.38%
Graph WaveNet	1.09	2.20	2.18%	1.32	3.01	2.80%	1.60	3.71	3.60%
ASTGCN	1.15	2.38	2.41%	1.42	3.23	3,28%	1.71	3.86	4.09%
DCRNN	1.13	2.30	2.31%	1.38	3.07	3.02%	1.70	3.94	3.94%
MTGNN	1.33	2.84	2.84%	1.65	3.63	3.55%	1.89	4.42	4.43%
AGCRN	1.35	2.85	2.94%	1.67	3.81	3.84%	1.96	4.57	4.69%
STASG(ours)	1.09	2.21	2.21%	1.32	2.94	2.83%	1.57	3.56	3.56%
STASG(2 layer)	1.08	2.18	2.18%	1.34	2.96	2.83%	1.57	3.60	3.52%

Table 3. Comparison of the performance of individual models on the METR-LA dataset.

Model	Timestep:15MIN			Timestep:30MIN			Timestep:60MIN		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
HistorialAverage	11.00	14.73	23.32%	11.00	14.73	23.32%	11.00	14.73	23.34%
LSTNet	3.80	8.06	9.16%	5.16	10.32	12.11%	6.12	12.01	15.00%
STGCN	3.43	7.89	8.50%	4.23	9.85	10.72%	5.18	11.96	12.90%
GMAN	3.24	6.93	8.79%	3.67	8.27	10.31%	4.34	9.67	12.82%
Graph WaveNet	2.83	6.50	6.73%	3.35	7.80	7.86%	4.57	10.14	12.21%
ASTGCN	3.10	7.04	7.53%	3.66	8.33	9.20%	4.34	9.46	11.34%
DCRNN	2.91	6.58	6.92%	3.31	7.79	8.02%	4.14	9.57	11.17%
MTGNN	3.30	6.85	7.17%	3.85	7.54	8.76%	4.00	9.15	10.28%
AGCRN	3.39	6.71	7.22%	3.92	7.45	8.78%	4.02	9.22	10.53%
STGAG(ours)	2.81	6.38	6.61%	3.30	7.70	7.79%	3.89	9.11	10.07%
STGAG(2 layer)	2.87	6.49	6.87%	3.30	7.73	7.90%	4.01	9.33	10.15%

We compared the performance of STASG with the benchmark models in Table 1 for 15 min (3 steps), 30 min (6 steps), and 60 min (12 steps) on the METR-LA and PEMS-BAY datasets.

We find that (1) our mod and other deep learning-based mods that take into account the graph structure are able to outperform the machine learning methods HistorialAverage, ARIMA. (2) GMAN and our mod also outperform traditional graph deep learning mods, which indicates the importance of capturing dynamic spatial-temporal correlations. (3) Compared with the benchmark, our mod achieves state-of-the-art prediction performance, and the advantage is more pronounced in long-term range prediction. The relatively low performance in short-term prediction may be due to the fact that the capture of temporal and spatial features is less effective in short-time prediction, while gated self-attention can play a more important role in long-time temporal feature capture, as longer sequences may contain more dependencies by which more local information of the data can be obtained.

4.5 Ablation Experiment

To evaluate the effect of key components that contribute to the improved results of our proposed mod, we designed six variants of STASG for ablation experiments on the METR-LA dataset. We named the variants of STASG as:
w/o no_GCN: On the basis of STASG, this method removes the simple graph convolution module.
w/o SGCN: On the basis of STASG, this method replaces the simple graph convolution module with the normal graph convolution module.
w/o STE: On the basis of STASG, the input of T-GAU removes the temporal embedding **matrix.**
w/o learn: On the basis of STASG, the adaptive matrix is removed from the self-learning matrix.
w/o GAU: On the basis of STASG, the gated attention unit is replaced with a normal self-attentive unit.
w/o MEAN_MAE: On the basis of STASG, this method replaces the training function MEAN_MAE with MAE.

5 Conclusion

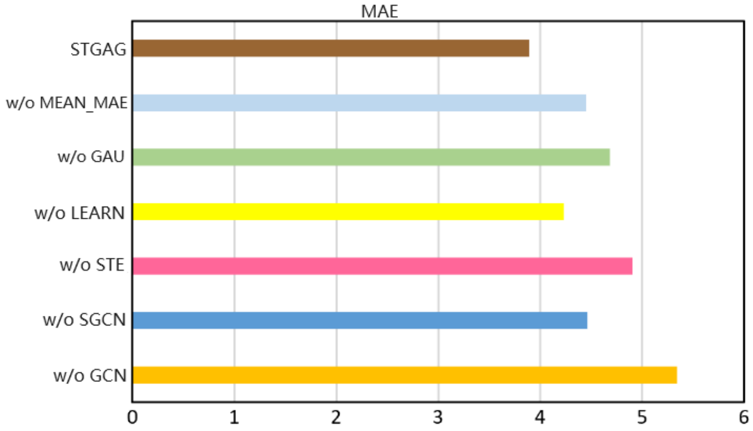
In this paper, we propose a novel spatio-temporal traffic prediction model called STASG based on gated self-attentiveness.

STASG is constructed using an encoder-decoder structure. Each encoder and decoder is constructed by stacking ST blocks, which not only capture the spatiotemporal structure of the input data and the adjacency matrix, but also fuse the two with a gating mechanism.

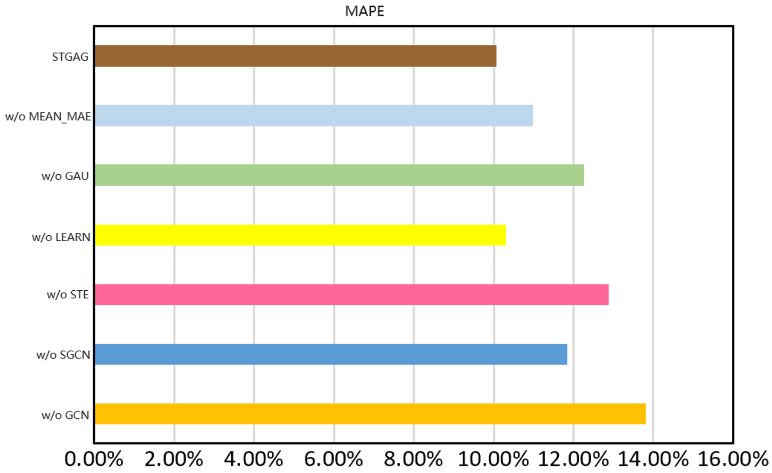
In addition to avoid the mean shift phenomenon in model prediction, we construct a new prediction loss function using mean deviation loss to limit the data range for inference of future data.

Several experiments and studies on two traffic flow datasets, PEMS-BAY and METR-LA, show that our model STASG performs better in traffic flow prediction. The comparison of the ablation experiments demonstrates the impact of the components of our model on the model prediction.

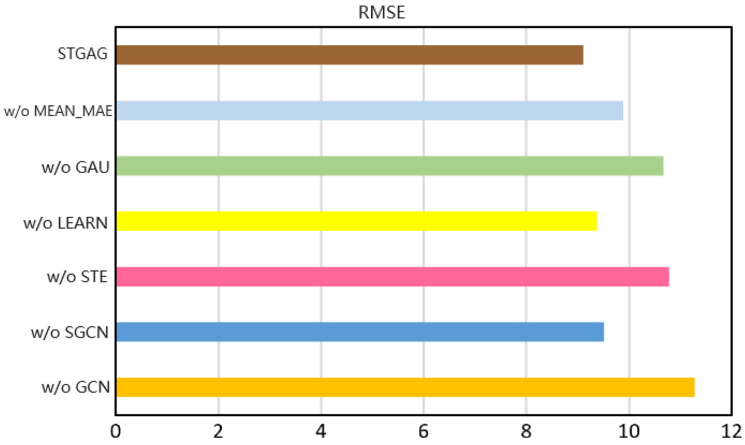
For future work, we plan to try to introduce other factors affecting traffic such as weather, holidays, and other time-varying variables into our feature fusion to improve the accuracy of the model.



(a)MAE in METR-LA dataset



(b) MAPE in METR-LA dataset



(c) RMSE in METR-LA dataset

Fig. 4. Performance metrics in METR-LA dataset

Acknowledgement. This work is funded in part by the National Natural Science Foundation of China Project (No. 62372078, 62376049) and the Dalian Key Field Innovation Team Support Plan (Grant: 2020RT07).

References

1. Alruban, A., Mengash, H. A., Eltahir, M. M., Almalki, N. S., Mahmud, A., Assiri, M.: Artificial Hummingbird Optimization Algorithm with Hierarchical Deep Learning for Traffic Management in Intelligent Transportation Systems. IEEE Access, 17596--17603 (2024)
2. Wang, K., Zhou, T., Li, X., Ren, F.: Performance and Challenges of 3D Object Detection Methods in Complex Scenes for Autonomous Driving. IEEE Transactions on Intelligent Vehicles 8(2), 1699-1716 (2023)
3. Yao, S., et al.: Radar-Camera Fusion for Object Detection and Semantic Segmen-tation in Autonomous Driving: A Comprehensive Review. IEEE Transactions on Intelligent Vehicles 9(1), 2094-2128 (2024)
4. Liu, M., Wang, W., Hu, X. et al.: Multivariate long-time series traffic passenger flow prediction using causal convolutional sparse self-attention MTS-Informer. Neural Comput & Applic 35, 24207–24223 (2023)
5. Rangapuram, S. S., Kapoor, S., Nirwan, R. S., Mercado, P., Januschowski, T., Wang,Y., & Bohlke-Schneider, M. Coherent Probabilistic Forecasting of Temporal Hierarchies. In: Proceedings of The 26th International Conference on Artificial Intelligenceand Statistics pp. 9362–9376 PMLR, Palau de Congressos, Valencia, Spain (2023)
6. Das, A., Kong, W., Paria, B., & Sen, R. Dirichlet Proportions Model for Hierarchically Coherent Probabilistic Forecasting. In: Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence pp. 518–528, PMLR, Pittsburgh, PA, USA (2023)
7. Yao, J.-P., Ling, Y., Hou, P., Wang, Z.-Y., & Huang, L. A graph neural network model for deciphering the biological mechanisms of plant electrical signal classification. Appl. Soft Comput., 137, 110153(2023)

8. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
9. Song, C., Lin, Y., Guo, S., et al.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 1, pp. 914–921 (2020)
10. Wu, Z., Pan, S., Long, G., et al.: Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 753–763.ACM (2020)
11. Rahman, R., Hasan, S.: Data-Driven Traffic Assignment: A Novel Approach for Learning Traffic Flow Patterns Using a Graph Convolutional Neural Network. IEEE Transactions on Intelligent Transportation Systems 23(1), 1–12 (2022)
12. Makridakis, S., Hibon, M.: ARMA models and the Box-Jenkins methodology. Journal of Forecasting 16(3), 147–163 (1997)
13. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1 (2017)
14. Ma, X., Tao, Z., Wang, Y., et al.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies 54, 187–197 (2015)
15. Nayak, N.: 5G Traffic Prediction with Time Series Analysis. Computational Intelligence in Internet of Things Enabled Applications, vol. 2022, 3174530 (2022)
16. Shi, X., Chen, Z., Wang, H., et al.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15), vol. 1, pp. 802–810. Springer, Montreal, Canada (2015)
17. Jiang, W., Zhang, L.: Geospatial data to images: A deep-learning framework for traffic forecasting. Tsinghua Science and Technology 24(1), 52–64 (2019)
18. Qu, Y., Rao, J., Gao, S., Zhang, Q., Chao, W.-L., Su, Y., Miller, M., Morales, A., Huber, P. R. FLEE-GNN: A Federated Learning System for Edge-Enhanced Graph Neural Network in Analyzing Geospatial Resilience of Multicommodity Food Flows. CoRR, abs/2310.13248. (2023)
19. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 Long Beach, CA, USA (2017)
20. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 (2018)
21. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving Language Understanding by Generative Pre-Training. arXiv:1706.03762 (2018)
22. Floridi, L., Chiriatti, M.: GPT-3: Its Nature, Scope, Limits, and Consequences. Minds and Machines 30, 681–694 (2020)
23. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv:11929 (2010)
24. Liu, Z., Lin, Y., Cao, Y., et al.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022, IEEE, Nashville, TN, USA (2021)
25. Liang, Y., Ke, S., Zhang, J., et al.: Geoman: Multi-Level Attention Networks for Geo-Sensory Time Series Prediction. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), pp. 3428–3434 AAAI Press, Stockholm, Sweden (2018)

26. Zheng, C., Fan, X., Wang, C., et al.: GMAN: A Graph Multi-Attention Network for Traffic Prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, pp. 1234–1241 AAAI, New York, USA (2020)
27. Ababio, I., Chen, J., Chen, Y., Xiao, L.: Link Prediction Based on Heuristics and Graph Attention. In: Proceedings of 2020 IEEE International Conference on Big Data (Big Data), 10-13 Online (2020)
28. Jiang, R., Yin, D., Wang, Z., et al.: DL-Traff: Survey and Benchmark of Deep Learning Models for Urban Traffic Prediction. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 4515– 4525 ACM, Gold Coast, Queensland, Australia (2021)
29. Yu, B., Yin, H., Zhu, Z.: Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In: Proceedings of Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18), pp. 3634– 3640, AAAI Press, Stockholm, Sweden (2018)
30. Lai, G., Chang, W.C., Yang, Y., et al.: Modeling Long and Short-Term Temporal Patterns with Deep Neural Networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18), pp. 95–104 ACM, Ann Arbor, Michigan, U.S.A. (2018)
31. Wu, Z., Pan, S., Long, G., et al.: Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19), pp. 1907–1913. AAAI Press, Macao, China (2019)
32. Guo, S., Lin, Y., Feng, N., et al.: Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 922–929 AAAI, Hawaii, USA (2019)
33. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv:1707.01926 (2017)
34. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. In: Advances in Neural Information Processing Systems, vol. 33, pp. 17804–17815 (2020)
35. Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 753–763, Virtual Event, CA, USA, (2020)
36. Shuvro, A. A., Khan, M. S., Rahman, M., Hussain, F., Moniruzzaman, M., & Hossen, M. S. Transformer Based Traffic Flow Forecasting in SDN-VANET. IEEE Access, 11, 41816–41826 (2023)
37. Huang, J., Zhao, P., Wang, G., Yang, S., & Lin, J. Self-attention-based long temporal sequence modeling method for temporal action detection. Neurocomputing, 554, 126617, (2023)