Cross-dynamic Spatial Temporal Transformer Network for Traffic Prediction

Anonymous Author
Anonymous Institution

Abstract

Accurate long-term traffic flow prediction is critical in building intelligent transportation systems. Although there are many existing works predicting traffic conditions in future time steps, most of them model dynamic spatial and temporal dependencies separately instead of a joint manner. To address this issue, we propose a novel traffic prediction network termed cross-dynamic spatial temporal transformer network(CSTTN), which adopts a transformer encoder-decoder architecture. Specifically, CSTTN captures the highly dynamic spatial-temporal correlations of traffic data jointly and integrally to augment the road nodes representations for accurate prediction. Furthermore, CSTTN utilizes the proposed cross-dynamic decoders to build the bridge between the spatial and temporal information extracted by encoders. Compared with previous works, CSTTN enables more efficient and scalable training for long-range spatial-temporal dependencies. Experimental results on two real-world public datasets, namely PeMS-BAY and PeMSD7, demonstrate that the proposed framework is competitive with the state-of-the-arts. The code is available at https://github.com/0shelter0/CSTTN.

1 INTORDUCTION

As an important part of Intelligent Transportation System (ITS)(Mori et al., 2015), real-time and accurate traffic prediction is of great significance to traffic resource management and can improve the efficiency of traffic networks.

Traffic forecasting can be regarded as a typical spatial-temporal graph data prediction problem. It is hoped to capture the dynamically spatial-temporal correlations between nodes from changing historical data to predict future traffic conditions (e.g., speeds, volumes and density). Therefore, how to effectively and efficiently model complex and dynamical spatial-temporal correlations from input data becomes the essential to accurate traffic prediction. Furthermore, the challenges of traffic forecasting are divided into the following two aspects:

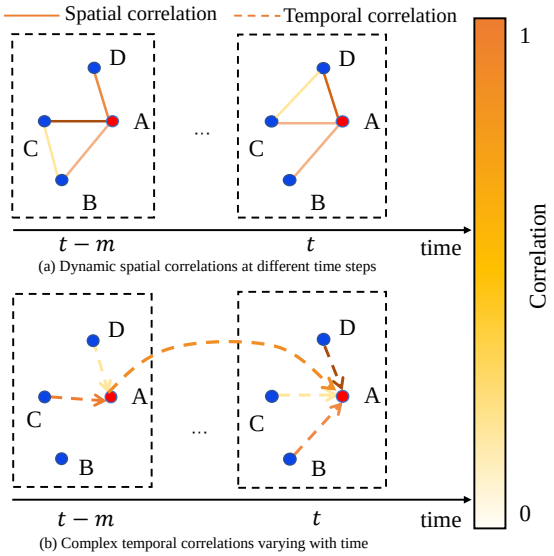


Figure 1: The dynamic spatial-temporal correlations diagram of traffic data.

1. *Dynamic Spatial Correlation.* In a traffic network, the target node is affected by its neighbors varying with time. As shown in Figure 1(a), the spatial correlation between target node A and its neighborhoods(node B, C and D) changes with time. For example, the correlation between node A and C is stronger at time step $t-m$ than that of at time step t, due to some external factors (e.g., speed, peak periods, weather conditions, etc.). Thus, effectively capturing this dynamic correlation is significant for accurate prediction.

2. *Complex Temporal Correlation.* The traffic status of each node is influenced by historical observations of its neighbors as well as itself, presenting complex and dynamic patterns. As shown in Figure 1(b), the traffic status

Preliminary work. Under review by AISTATS 2023. Do not distribute.

of node A will be affected by itself and its neighbors, and this influence will be different at time step $t-m$ and t.

In the early deep learning methods, convolutional neural networks(CNNs) are frequently used to extract spatial dependencies, while recurrent neural networks(RNNs) are widely applied to capture temporal dependencies. However, CNNs can not extract the topology information of road network well. Recently, Graph Neural Networks(GNNs)(Kipf and Welling, 2016; Defferrard et al., 2016; Bruna et al., 2013) have been proposed as generalization of convolution in non-Euclidean space, which can extract inherent topology information of graph.

In order to capture spatial correlations, GNNs have recently been widely integrated into various RNN-based sequence models(Li et al., 2017; Zhao et al., 2019) and have achieved improvements. With the advent of Transformer(Vaswani et al., 2017), there have been many models with transformer-based encoder and decoder architecture for traffic forecasting. The proposed self-attention(Vaswani et al., 2017) mechanism is adopted to model dynamic spatial-temporal correlations by these approaches(Zheng et al., 2020; Guo et al., 2021; Xu et al., 2020; Guo et al., 2019), which achieved promising performance compared to previous methods especially in long-term prediction.

For the above models based Transformer, some merely utilize the encoder part, while others stack their spatial attention and temporal attention modules directly or use parallel modules and then fuse them simply, still others execute prediction in an auto-regressive way. They do not exploit the spatial-temporal features simultaneously and complementary. Instead, the spatial attention module is first applied to extract spatial dependencies, and then as inputs of the temporal attention module, the spatial-temporal features are obtained. However, this separately spatial-temporal modeling approach may not adequately incorporate the spatial and temporal correlations and contextual information.

Therefore, we propose an trainable end-to-end framework termed Cross-dynamic Spatial Temporal Transformer Network(CSTTN), which models spatial-temporal correlations jointly to address the aforementioned challenges in a cross-dynamic manner. Specifically, we utilize encoders to embed spatial and temporal correlations to obtain features, and then adopt decoders that interactively build a bridge between the embedded spatial and temporal features to capture the comprehensive representations for prediction. By stacking multiple CSTT blocks, more refined and deep spatial-temporal features can be learned.

The contributions of this paper are summarized as follows:

- We propose a transformer-based architecture that jointly models the spatial and temporal correlations of traffic data in a cross-dynamic manner.
- We use gated mechanism to fuse outputs of the spatial-temporal decoders and make prediction in a non-autoregressive way.
- We conduct extensive experiments on publicly available datasets and achieve results competitive with the state-of-the-arts.

2 RELATED WORKS

We first review the existing approaches for traffic prediction , and then briefly introduce the use of attention mechanism to model spatial and temporal correlations.

Traffic Prediction Traffic prediction has been extensively studied over the past decades. The early data-driven approaches fall into two major categories: traditional time series models and machine learning models. Auto-Regressive Integrated Moving Average (ARIMA)(Makridakis and Hibon, 1997) is a representative of time series models, fitting on observed time series to predict future data. Linear regression models and Kalman filter models are general methods within the machine learning models category. Linear regression models build a regression function based on historical observations, and then the traffic forecasting realization is based on the regression function. Typical approaches include Vector Auto-Regressions (VAR)(Lippi et al., 2013) which estimates the relationship between the time series and their lagged values, and Support Vector Regression (SVR)(Wu et al., 2004) which uses a support vector machine to do regression on the traffic sequence. The Kalman filter models(Kumar, 2017; Ojeda et al., 2013) predict future traffic conditions based on the traffic state of the previous moment and the current moment. Although these traditional methods have simple algorithms and fast calculation, they cannot take spatial or long-term dependence into account and perform with low prediction precision. Since the above approaches rely on the ideal stationary assumption about time series, cannot reflect the complicated nonlinearities and dynamics of traffic data, and cannot overcome the interference of external factors such as traffic accidents and weather conditions.

With the rapid development of deep learning, RNN and its variants(LSTM and GRU) are widely applied to extract temporal information from sequences(e.g., FC-LSTM(Sutskever et al., 2014)), but can not take the spatial correlation of traffic data into account. Since traffic data can be represented as time series on the road network, many models based on GNNs have been proposed to extract both temporal and spatial correlations, such as T-GCN(Zhao et al., 2019) integrates graph convolution and GRU, DCRNN(Li et al., 2017) uses diffusion graph convolution to extract spatial dependencies combined with RNN to extract temporal dependencies, and Graph WaveNet(Wu et al., 2019) utilizes diffusion graph convolution and 1D dilated casual convolution. However, these methods use

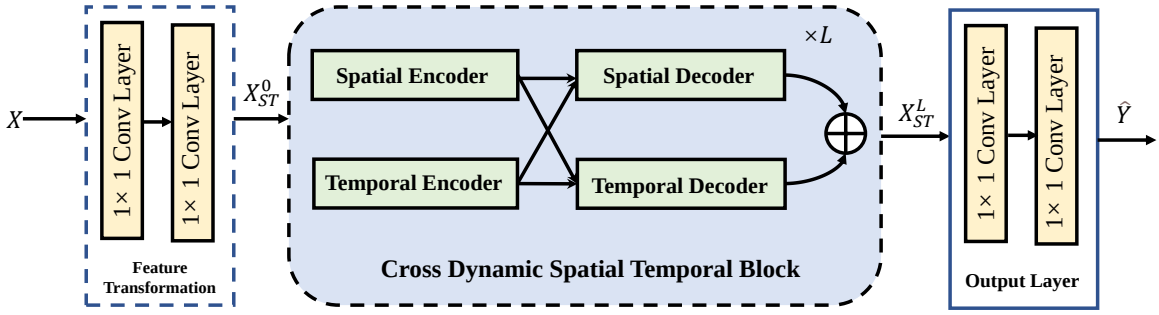


Figure 2: Illustration of the proposed cross-dynamic spatial temporal transformer network(CSTTN), which consists of a features embedding layer, multiple stacked cross-dynamic spatial temporal(CST) blocks and an output layer. Each of CST block contains encoders and decoders built in a cross manner over spatial and temporal dimensions. The architecture generates prediction results in a multi-steps way.

a predefined graph structure, and the spatial dependencies will be fixed during the training process, which makes it impossible to mine potential dynamic spatial correlations. In this paper, the transformer-based encoders are applied to extract long-range temporal dependencies and capture dynamic spatial correlations, instead of only using the static spatial connectivity information of road network.

Spatial-Temporal Transformer Transformer was originally proposed in Vaswani et al. (2017) for tasks in natural language processing, such as machine translation, abstract summarization and textual entailment. Recently, Transformers have been extensively applied in computer vision tasks(e.g., image classification(Dosovitskiy et al., 2020), object detection(Carion et al., 2020), video action recognition(Bertasius et al., 2021)) and achieved great results. The proposed self-attention mechanism is suitable for dealing with sequence-to-sequence tasks by modeling long-range dependencies and can speed up training with more parallelization. In the field of traffic prediction, many recent approaches adopt attention mechanism to extract spatial-temporal features and achieve good results, especially in long-range scales. GMAN(Zheng et al., 2020) proposes the ST-Attention block, which models dynamic spatial-temporal correlations in both encoder and decoder, and adds a transform attention layer between encoder and decoder. ASTGCN(Guo et al., 2019) uses attention mechanism in spatial and temporal dimensions respectively, learning spatial-temporal features. STTN(Xu et al., 2020) designs a general dynamic graph neural network to model the spatial dependencies evolving over time. Additionally, There are some approaches to apply attention mechanisms in domain of computer vision, such as TimeSformer(Bertasius et al., 2021) utilizes self-attention to model video data from space-time dimensions separately, which extends ViT(Dosovitskiy et al., 2020) in video understanding; Existing methods that model spatial-temporal correlations based on attention mechanism can be divided into following categories: capture spatial and temporal correlations in a stacked manner, or in parallel then fused them simply. Unlike the above approaches, we first embed the spatial and temporal correlations in parallel, then employ a decoder to complementary exploit spatial-temporal correlations by means of a cross-dynamic way following GroupFormer(Li et al., 2021).

3 METHODOLOGY

In this section, the mathematical definition of problem we focus on will be first given. Next, we overview the architecture of the proposed cross-dynamic spatial temporal transformer network. Finally, we introduce each component of CST block in detail.

3.1 Problem Definition

In this study, a traffic network is denoted as a directed graph $G = (V, E, \mathrm{A})$, where V is a set of $N = |V|$ nodes representing sensors in traffic network; E is a set of edges indicating the connectivity among the nodes; and $\mathrm{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix of graph G representing the proximity measured by the Euclidean distances between sensors via Gaussian kernel.

At each time step t, the traffic conditions can be represented as a graph signal $X_t \in \mathbb{R}^{N \times C}$ on graph G, where C is the number of traffic conditions observed such as traffic speed, traffic volume, traffic density and so on.

Given a traffic network graph G and traffic conditions of historical P time steps $[X_{t-P+1}, \cdots, X_t] \in \mathbb{R}^{P \times N \times C}$ observed by the N nodes, we aim to learn a function f to predict the traffic conditions of the next Q time steps over all nodes. The mapping function can be written as:

$$\hat{X}_{t+1}, \cdots, \hat{X}_{t+Q} = f(X_{t-P+1}, \cdots, X_t; G) \quad (1)$$

where $[\hat{X}_{t+1}, \cdots, \hat{X}_{t+Q}] \in \mathbb{R}^{Q \times N \times C}$.

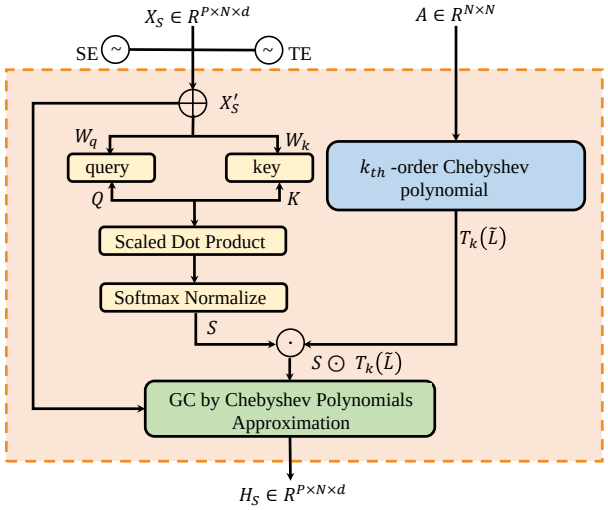


Figure 3: The spatial encoder models static and dynamic spatial correlations using attention mechanism and graph convolution approximated by Chebyshev Polynomials. SE and TE is spatial and temporal positional embedding respectively, $\oplus$ denotes the addition operation, and $\odot$ denotes element-wise product.

3.2 Overview

In this section, we elaborate on the proposed architecture of cross-dynamic spatial temporal transformer network(CSTTN). As illustrated in Figure 2, CSTTN is mainly composed of several cross-dynamic spatial temporal(CST) blocks. Each of CST block includes two encoders(i.e., a spatial encoder and temporal encoder) in parallel to extract spatial and temporal features respectively. Subsequently, the decoders are developed to decode the spatial-temporal correlations with a cross-dynamic manner. Multiple CST blocks can be stacked to model the deep spatial-temporal correlations from the traffic data of inputs. Finally, the extracted spatial-temporal features are aggregated by an output layer to generate the prediction results.

3.3 Cross-Dynamic Spatial Temporal Blocks

In proposed cross-dynamic spatial temporal(CST) block, we utilize encoders and decoders of Transformer(Vaswani et al., 2017) to integrally model dynamic spatial-temporal correlations for accurate traffic prediction. The details of each component will be described as follows.

3.3.1 Spatial Temporal Encoders

The traffic conditions of future time steps at a location are affected by the historical observations of itself and its neighborhoods with different impacts. Such impact is highly dynamic and complex, varying with time. To this end, two parallel encoders are deployed to embed spatial and temporal contextual features.

Spatial Encoders Given observations $X \in \mathbb{R}^{P\times N\times C}$ over historical P time steps for N nodes in traffic network, we first embed X into a d-dimensionality space to get $X' \in \mathbb{R}^{P\times N\times d}$ by employing feature embedding layer which contains two 1×1 convolution layers. Since the attention mechanism adopts a fully-connected structure, it cannot model the positional information in sequence. Therefore, before X' fed into the spatial encoder, we adopt learnable spatial and temporal positional embedding referred to Xu et al. (2020). Specifically, the spatial topology structure of road network and time step information will be learned as embeddings $E_S \in \mathbb{R}^{N\times N}$ and $E_T \in \mathbb{R}^{P\times P}$ respectively, where E_S is initialized with the adjacency matrix of G and E_T is initialized with one-hot encoding of P historical time steps. Next, with two fully-connected layers E_S and E_T are transformed to tensors $\mathbb{R}^{N\times d}$ and $\mathbb{R}^{P\times d}$, both of which add with X' to obtain the position-embedded features $X_S \in \mathbb{R}^{P\times N\times d}$ by broadcasting mechanism.

As shown in Figure 3, we propose a spatial encoder that integrates attention mechanism and graph convolution based on Chebyshev polynomials approximation for taking static dependencies in graph and dynamic spatial correlations between nodes into account. Specifically, we denote the input of spatial encoder in l^{th} CST block as $X_{ST}^{(l-1)} \in \mathbb{R}^{P\times N\times d}$, with $X_{ST}^{(0)} = X_S, l = 1, \cdots, L$. We view the temporal dimension of $X_{ST}^{(l-1)}$ as the batch dimension and adopt attention mechanism to calculate the spatial correlations between nodes over P time steps, expressed as follows:

$$Q_{EN_S} = X_{ST}^{(l-1)} W_{EN_S}^q,\ K_{EN_S} = X_{ST}^{(l-1)} W_{EN_S}^k,$$
$$S = \text{softmax}\left(\frac{1}{\sqrt{d}} Q_{EN_S} K_{EN_S}^T\right), \tag{2}$$

where $W_{EN_S}^q, W_{EN_S}^k \in \mathbb{R}^{d\times d}$ are learnable projection matrices shared by all nodes, $S \in \mathbb{R}^{P\times N\times N}$ denotes the spatial correlation matrix for all nodes over all time steps.

For capturing the static spatial dependencies of input data in road network, we apply graph convolution through Chebyshev polynomials approximation(Defferrard et al., 2016). Specifically, we denote the normalized Laplacian matrix of graph G as $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \in \mathbb{R}^{N\times N}$, where A is the adjacent matrix, I_N is an identity matrix, and the diagonal degree matrix $D \in \mathbb{R}^{N\times N}$ consists of node degrees $D_{ii} = \sum_j A_{ij}$. The graph convolution can then be formulated as:

$$\mathbf{H}_S^{(l)} = \Theta(L) X_{ST}^{(l-1)} \approx \sum_{k=0}^{K-1} \Theta_k T_k(\tilde{L}) X_{ST}^{(l-1)}, \tag{3}$$

where $\Theta \in \mathbb{R}^{d\times d}$ is the vector of polynomial coefficients to be learned. $T_k(\tilde{L}) \in \mathbb{R}^{N\times N}$ is the k^{th}-order Chebyshev polynomial taking the scaled Laplacian matrix $\tilde{L} =$

$2L/\lambda_{max} - I_N$ as input, and λ_{max} is the maximum eigenvalue of L. Through the graph convolution operation in Eq.(3) each node feature can be extracted by aggregating information of the 0 to $(K-1)^{th}$-order neighbors centered on itself in the graph.

To dynamically model the spatial correlations between nodes, we combine $T_k(\tilde{L}) \in \mathbb{R}^{N\times N}$ with the spatial correlation matrix $S \in \mathbb{R}^{P\times N\times N}$ via broadcasting mechanism, then obtain $T_k(\tilde{L}) \odot S$, here $\odot$ denotes the element-wise product. Thus, the graph convolution in Eq.(3) can be rewritten as:

$$\mathbf{H}_S^{(l)} = \Theta(L)X_{ST}^{(l-1)} \approx \sum_{k=0}^{K-1} \Theta_k(T_k(\tilde{L}) \odot S)X_{ST}^{(l-1)}, \quad (4)$$

where $\mathbf{H}_S^{(l)} \in \mathbb{R}^{P\times N\times d}$ is output of the l^{th} CST block.

Temporal Encoders The other parallel temporal encoder is employed to augment the input features with long-range temporal correlations by highlighting relative evolution clues. The temporal encoder in l^{th} CST block takes $X_{ST}^{(l-1)}$ as input, where $X_{ST}^{(0)} = X_T \in \mathbb{R}^{P\times N\times d}$ generated from the temporal positional embedding. Similar to spatial encoder, the temporal encoder views the spatial dimension of $X_{ST}^{(l-1)}$ as a batch dimension. For simplicity, We take the features $X_{ST,v_i}^{(l-1)} \in X_{ST}^{(l-1)}$ shaped as $P \times d$ of one arbitrary node v_i in G for illustration. The process of embedding temporal correlations for node v_i by self-attention mechanism can be denoted as:

$$\begin{aligned} Q_{EN_T} &= X_{ST,v_i}^{(l-1)}W_{EN_T}^q, K_{EN_T} = X_{ST,v_i}^{(l-1)}W_{EN_T}^k, \\ V_{EN_T} &= X_{ST,v_i}^{(l-1)}W_{EN_T}^v, \\ Z_{EN} &= \mathrm{softmax}(\frac{Q_{EN_T}K_{EN_T}^T}{\sqrt{d}})V_{EN_T} + V_{EN_T}, \\ \mathbf{H}_{T,v_i}^{\prime(l)} &= \mathrm{ReLU}(Z_{EN}W_0)W_1, \end{aligned} \quad (5)$$

where $W_{EN_T}^q W_{EN_T}^k, W_{EN_T}^v \in \mathbb{R}^{d\times d}$ are learnable parameters for linear projection, $W_0, W_1 \in \mathbb{R}^{d\times d}$ denotes parameter matrices of feed-forward network in canonical Transformer, and $\mathrm{ReLU}(\cdot)$ denotes the nonlinear activation function. The embedded features for all nodes $\{\mathbf{H}_{T,v_i}^{\prime(l)} \in \mathbb{R}^{P\times d} | \forall v_i \in V\}$ are packed together into $\mathbf{H}_T^{\prime(l)} \in \mathbb{R}^{P\times N\times d}$ along spatial axis. Residual connection $\mathbf{H}_T^{(l)} = \mathbf{H}_T^{\prime(l)} + X_{ST}^{(l-1)}$ is applied for stable training.

3.3.2 Spatial Temporal Decoders

The spatial and temporal decoders are utilized to connect the spatial and temporal correlations mutually. The decoders following the standard architecture of Transformer are deployed to complementary exploit spatial-temporal correlations through a cross-dynamic way.

Decoders The spatial decoder takes $\mathbf{H}_S^{(l)}$ as the *node query*, these queries of N nodes over each time step are transformed into embedded features with abundant spatial-temporal semantic correlations by decoder, where the temporal embedding $\mathbf{H}_T^{(l)}$ is viewed as *key* and *value*. The *node query* captures temporal dynamic correlations from *key*, obtaining the updated temporal-context features $V_{ST}^{(l)} \in \mathbb{R}^{P\times N\times d}$. We view the temporal dimension of *query, key* and *value* as a batch dimension similar to the previous, and the output $V_{ST}^{(l)}$ of the spatial decoder in the l^{th} CST block can be updated as follows:

$$\begin{aligned} Q_{DE_S} &= \mathbf{H}_S^{(l)}W_{DE_S}^q, K_{DE_S} = \mathbf{H}_T^{(l)}W_{DE_S}^k, \\ V_{DE_S} &= \mathbf{H}_T^{(l)}W_{DE_S}^v, \\ Z_{DE} &= \mathrm{softmax}(\frac{Q_{DE_S}K_{DE_S}^T}{\sqrt{d}})V_{DE_S} + V_{DE_S}, \\ V_{ST}^{(l)} &= \mathrm{ReLU}(Z_{DE}U_0)U_1 \end{aligned} \quad (6)$$

where $W_{DE_S}^q, W_{DE_S}^k, W_{DE_S}^v, U_0, U_1 \in \mathbb{R}^{d\times d}$ are all weight matrices to be learned.

Simultaneously, we also apply the other temporal decoder by a cross-dynamic manner. Specifically, the spatial embedding $\mathbf{H}_S^{(l)}$ extracted by spatial encoder transposes the temporal dimension with spatial dimension and can be regarded as the *key* and *value* adopted by the decoder. The decoder views temporal embedding $\mathbf{H}_T^{(l)}$ as *time-step query* and extracts spatially embedded features $V_{TS}^{(l)} \in \mathbb{R}^{P\times N\times d}$. In this process, the *key* represents spatial features $\mathbf{H}_S^{(l)}$ along with time order and *time-step query* of each node in G explores the time steps of interest in the historical traffic data. The purpose of spatial and temporal decoders designed by cross-dynamic scheme is to exploit semantic associations based on the spatial correlation and temporal correlation for enhancing representation of each node.

Gated Fusion for Decoders Finally, the output embeddings of the spatial and temporal decoders are adaptively fused via a gated mechanism to generate the augmented node representation $Y_{ST}^{(l)} \in \mathbb{R}^{P\times N\times d}$. In the l^{th} block, the outputs of the spatial and temporal decoders are represented as $V_{ST}^{(l)}$ and $V_{TS}^{(l)}$ both shaped as $\mathbb{R}^{P\times N\times d}$. The gate g is derived from $V_{ST}^{(l)}$ and $V_{TS}^{(l)}$ as:

$$g = \sigma(V_{ST}^{(l)}W_{g1} + V_{TS}^{(l)}W_{g2} + b_g), \quad (7)$$

where $W_{g1}, W_{g2} \in \mathbb{R}^{d\times d}$ and $b_g \in \mathbb{R}^d$ are learnable parameters, $\sigma(\cdot)$ denotes the sigmoid activation function. Furthermore, the output $Y_{ST}^{(l)} \in \mathbb{R}^{P\times N\times d}$ of gated fusion mechanism is generated by weighted summation of $V_{ST}^{(l)}$ and $V_{TS}^{(l)}$ with gate g.

$$Y_{ST}^{(l)} = g \odot V_{ST}^{(l)} + (1-g) \odot V_{TS}^{(l)}. \quad (8)$$

where $\odot$ represents the element-wise product. The flow of spatial and temporal correlations can be adaptively controlled by the gated fusion mechanism for each node and time step.

3.4 Framework Optimization

The output of the last block is denoted as $Y_{ST}^{(L)} \in \mathbb{R}^{P\times N\times d}$ and as input of the following output layer. The output layer includes two 1×1 convolution layers transforming the number of time steps and channels to predict, then obtaining the final prediction results $\hat{Y} \in \mathbb{R}^{Q\times N\times C}$. Our CSTTN framework can be trained in an end-to-end fashion. We choose to minimize the *mean absolute error*(MAE) loss between predicted values and ground truths to optimize network:

$$\mathcal{L}(\Theta) = \sum_t \|Y_t - \hat{Y}_t\| + \lambda L_{reg} \tag{9}$$

where Θ is all learnable parameters in CSTTN, and Y_t denotes the ground truth. The second term L_{reg} denotes the L2 regularization term that helps to avoid an overfitting problem during the training process and λ is a hyperparameter.

4 RESULTS

4.1 Datasets

Datasets description We evaluate our CSTTN on two public real-world traffic datasets, namely **PeMSD7** and **PEMS-BAY**, as detailed following. PeMSD7 records 2-month traffic data on 228 sensors over the weekdays of May and June of 2012 in District 7 of California. PEMS-BAY contains six months of traffic data on 325 sensors from January 1st 2017 to May 31th 2017 in the Bay area. PeMSD7 uses the first 34 days as training set, and the remaining as validation and test set following Yu et al. (2017). PEMS-BAY is divided into a training set(70%), validation set(10%), and test set(20%) in chronological order as Li et al. (2017). Statistics of the datasets are summarized in Table 1.

Datasets	#Nodes	#Edges	#TimeSteps	TimeSpan
PeMSD7	228	832	12672	2 month
PEMS-BAY	325	2369	52116	6 month

Table 1: Statistics of PeMSD7 and PEMS-BAY.

Datasets preprocessing We adopt the same data preprocessing procedures as in DCRNN(Li et al., 2017). In both datasets, the raw traffic data is aggregated into 5-minute interval corresponding to a time step. Therefore, every sensor contains 288 traffic data records per day. Z-score normalization is applied to preprocess the input data. To construct the traffic network graph G, each traffic sensor is considered as a node. Then, the adjacency matrix A of the nodes is generated by traffic network distance with a thresholded Gaussian kernel as follows:

$$A_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\sigma^2}), & \text{if } \exp(-\frac{d_{ij}^2}{\sigma^2}) \geq \epsilon, \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where d_{ij} stands for the traffic network distance from sensor i to j, σ is the standard deviation to control the distribution of A, and ϵ (assigned to 0.1) is the threshold to control the sparsity of A.

It is worth noting that the road network graph is predefined as directed in PEMS-BAY. Because there exist some pairs of nodes with different shortest topological distances in the actual road network between different directions. However, we expect to obtain symmetric adjacency matrix for Laplacian construction. To this end, we generate undirected graphs by selecting the larger weight from the two directions of each pair of nodes to construct symmetric adjacency matrix. In PeMSD7, the adjacency matrix is symmetric via computing the pairwise Euclidean distances between nodes. Using directed graph to extract spatial features will be left for future work.

4.2 Experimental Settings

Metrics We use three metrics to evaluate the performance of our model, i.e., Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

Implementation details Our experiments are conducted on one NVIDIA Geforce RTX 3090 card. Following previous works, we use observations over $P = 12$ historical time steps(1 hour) to predict the traffic conditions in the next $Q = 3$, 6, and 12 time steps. The proposed CSTTN is trained with Adam optimizer for 30 epochs with a batch size of 32. The initial learning rate is set to 0.001 with a decay rate of 0.5 after every 8 epochs. The number of traffic conditions on both datasets is $C = 1$. The graph convolution kernel size K is set to 3. We set the number of CST blocks L to 2, the number of hidden layers in encoders and decoders to 2 and 1 respectively, and the number of attention heads to 4 everywhere used. The dimensionality d of each CST block is set to 64. The L2 regularization term weight λ is set to 0.0001.

4.3 Baselines

We compare CSTTN with the following baselines:

- ARIMA(Makridakis and Hibon, 1997): Auto-Regressive Integrated Moving Average model with Kalman filter.

Dataset	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PEMS-BAY	ARIMA (Makridakis and Hibon, 1997)	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM (Sutskever et al., 2014)	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	DCRNN (Li et al., 2017)	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	STGCN (Yu et al., 2017)	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	T-GCN (Zhao et al., 2019)	1.50	2.83	3.14%	1.73	3.40	3.76%	2.18	4.35	4.94%
	STTN (Xu et al., 2020)	1.36	2.87	2.89%	1.67	3.79	3.78%	1.95	4.50	4.58%
	Graph WaveNet (Wu et al., 2019)	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	GMAN(Zheng et al., 2020)	1.34	2.82	2.81%	1.62	3.72	3.63%	1.86	4.32	4.31%
	CSTTN(Ours)	**1.33**	**2.81**	**2.79%**	**1.64**	**3.72**	**3.66%**	**1.95**	**4.47**	**4.59%**
PeMSD7	ARIMA (Makridakis and Hibon, 1997)	5.57	9.00	13.04%	5.94	9.22	14.01%	6.68	9.68	16.78%
	DCRNN (Li et al., 2017)	2.22	4.25	5.16%	3.04	6.02	7.46%	4.15	8.20	10.82%
	STGCN (Yu et al., 2017)	2.24	4.01	5.28%	3.04	5.74	7.46%	4.08	7.69	10.23%
	ASTGCN (Guo et al., 2019)	2.85	5.15	7.25%	3.35	6.12	8.67%	3.96	7.20	10.53%
	LSGCN (Huang et al., 2020)	2.22	3.98	5.14%	2.96	5.47	7.18%	3.81	7.09	9.60%
	STTN (Xu et al., 2020)	2.14	4.04	5.05%	2.70	5.37	6.68%	-	-	-
	Graph WaveNet (Wu et al., 2019)	2.14	4.01	4.93%	2.80	5.48	6.89%	-	-	-
	CSTTN(Ours)	**2.16**	**4.02**	**5.13%**	**2.76**	**5.33**	**6.82%**	**3.34**	**6.52**	**8.44%**

Table 2: Traffic prediction performance comparison of CSTTN and other baseline models on PEMS-BAY and PeMSD7.

- FC-LSTM(Sutskever et al., 2014): A sequence-to-sequence model, which uses fully-connected LSTM layers in both encoder and decoder.
- DCRNN(Li et al., 2017): Diffusion convolutional recurrent neural network, which combines graph diffusion convolution with recurrent neural networks in an encoder-decoder manner.
- STGCN(Yu et al., 2017): Spatial-temporal graph convolutional network, which combines graph convolution with standard 1D convolution in temporal dimension.
- T-GCN(Zhao et al., 2019): A temporal graph convolutional network, it simply combines the graph convolutional network and the gated recurrent unit for traffic prediction.
- GMAN(Zheng et al., 2020):A graph multi-attention network with spatial and temporal attention modules, employing encoder-decoder architecture.
- ASTGCN(Guo et al., 2019): Attention based spatial-temporal graph convolutional network, which adopts attention mechanism in spatial and temporal dimension to capture dynamics, then integrates graph convolution.
- STTN(Xu et al., 2020): Spatial-Temporal transformer network, which uses spatial and temporal transformer to model spatial-temporal correlations, but only uses encoders.
- Graph WaveNet(Wu et al., 2019): A model utilizes diffusion graph convolution and 1D dilated casual convolution.
- LSGCN(Huang et al., 2020): Long short-term graph convolutional network, which integrates graph convolution network(GCN) and the proposed graph attention network cosAtt to capture spatial features, and uses gated linear units convolution(GLU) to capture temporal features.

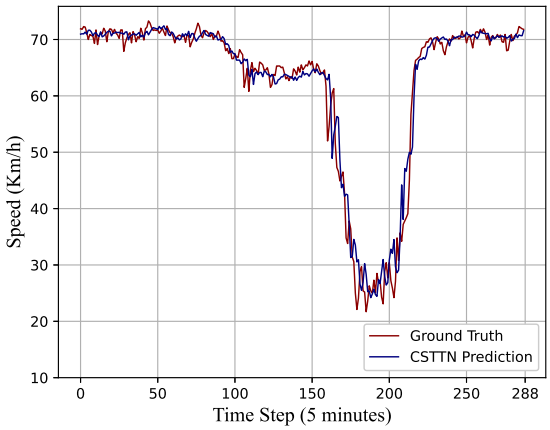
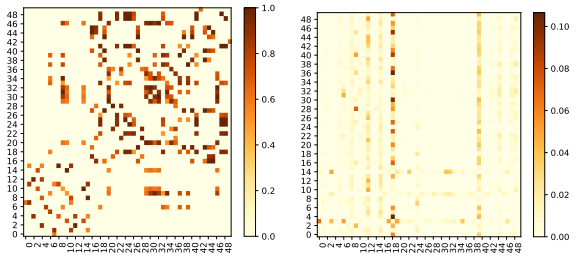


Figure 4: Visualization of one-day prediction results obtained by CSTTN on PeMSD7.

4.4 Experimental Results

Table 2 presents the experimental results of CSTTN and baselines for 15, 30 and 60 minutes ahead prediction on PeMSD7 and PEMS-BAY datasets. The prediction performance comparison shows that CSTTN can achieve competitive results with the state-of-the-art methods in short-term and long-term range on both datasets, while outperforming spatial-temporal models based on the predefined graph (STGCN(Yu et al., 2017), T-GCN(Zhao et al., 2019) and DCRNN(Li et al., 2017)).

For PEMS-BAY, CSTTN performs better than STTN and GMAN(Zheng et al., 2020) in short-term prediction ($\leq$30 min), and not as well as Graph WaveNet(Wu et al., 2019) which utilizes separate graph convolutional layers more suitable for focusing on localized time horizons. Moreover,



(a) Adjacency Matrix. (b) Learned spatial correlation matrix.

Figure 5: The heatmaps of the predefined adjacency matrix and the learned spatial correlation matrix for the first 50 nodes on PeMSD7, and (b) is generated by spatial encoder at time step 12.

Methods	CSTTN	w/o ST-PE	w/o Fusion	w/o SD	w/o TD
MAE	**3.34**	3.54	3.37	3.36	3.53
RMSE	**6.52**	6.87	6.61	6.58	6.78
MAPE	**8.44%**	9.05%	8.53%	8.50%	8.98%

Table 3: Ablation study on different variants.

CSTTN surpasses Graph WaveNet in long-term prediction (60 min), competitive with STTN(Xu et al., 2020), and not as well as GMAN, since GAMN employs a transform attention layer to effectively ease the effect of error propagation. For PeMSD7, CSTTN also presents the similar results. Additionally, CSTTN outperforms ASTGCN(Guo et al., 2019) and LSGCN(Huang et al., 2020) by a considerable margin. This is because that ASTGCN stacks spatial attention after temporal attention to model spatial-temporal correlations with some degree of limitation, and that LSGCN exploits spatial features on the static predefined graph, lacking of dynamic change over time.

The fact indicates that CSTTN performs better in long-term prediction, due to applying the cross-dynamic attention scheme to capture long-range temporal dependencies and alleviate the accumulation of error over time. Compared with STTN, the performance on the two datasets is somewhat different, but CSTTN is generally more stable. Because STTN only adopts encoders in a stacked manner, the integration of spatial and temporal correlations is not sufficient. However, CSTTN uses encoder-decoder architecture, and the decoder aggregates spatial and temporal correlations with a cross-dynamic manner, resulting in more semantic contextual representations of nodes.

4.5 Ablation Studies

To verify the effectiveness of each component in our model, we evaluate four variants as follows: (1) *w/o ST-PE*: CSTTN without the spatial and temporal positional embedding. (2) *w/o Fusion*: in this variant, the outputs of spatial decoder and temporal decoder are fused by summation directly instead of applying gated fusion mechanism. (3) *w/o SD*: in this variant, the spatial decoder is removed and the output layer only take output of the temporal decoder as input. (4) *w/o TD*: CSTTN without temporal decoder. Apart from the component removed of CSTTN, the other settings are all the same. We report the results of MAE, RMSE, MAPE in 60 minutes ahead prediction on PeMSD7 in Table 3. It is easily observed that CSTTN consistently outperforms the four variants, indicating the importance of these components. Compared to w/o TD, w/o SD performs better because the temporal decoder extracts temporal dependencies from spatial features arranged in time axis, and aggregate spatial and temporal features more refined than the spatial decoder.

4.6 Visualization

In this subsection, we visualize the results to illustrate the model's capacity for handling complex traffic situations. Specifically, we randomly sample a sensor node from PeMSD7 and then carry out a one-day prediction of traffic speed over 15-minute ahead horizon. As shown in Figure 4, the model can almost simultaneously fit the ground truth, and accurately detect the start and the end of the evening rush hours (around the 180 time step). This verifies that CSTTN can adequately model temporal correlations. Moreover, we plot the heatmap of the static adjacency matrix and the learned spatial correlation matrix on PeMSD7 in Figure 5. It can be observed that Figure 5b is more sparse and includes some latent spatial correlations between nodes compared to Figure 5a. The spatial correlation matrix is dynamically adjusted over time to extract spatial features more steadily and it overcomes the drawback that the static adjacency matrix only depends on the spatial distance.

5 CONCLUSIONS

In this paper, we propose a novel transformer-based architecture named CSTTN for traffic prediction, which utilizes both encoder and decoder in the Transformer and makes multi-step predictions by non-autoregressive means. Experimental results on two real-world datasets show that CSTTN can compete with the state-of-the-art methods, demonstrating its effectiveness on modeling spatial-temporal correlations from input data. Furthermore, we will apply our CSTTN into more general spatial-temporal prediction tasks.

References

Usue Mori, Alexander Mendiburu, Maite Álvarez, and Jose A Lozano. A review of travel time estimation and forecasting for advanced traveller information systems. *Transportmetrica A: Transport Science*, 11(2):119–157, 2015. 1

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 2, 4

J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networks and locally connected networks on graphs. *Computer Science*, 2013. 2

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017. 2, 6, 7

L. Zhao, Y. Song, C. Zhang, Y. Liu, and H. Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–11, 2019. 2, 7

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1234–1241, 2020. 2, 3, 7

Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 2

Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*, 2020. 2, 3, 4, 7, 8

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019. 2, 3, 7, 8

Spyros Makridakis and Michele Hibon. Arma models and the box–jenkins methodology. *Journal of forecasting*, 16 (3):147–163, 1997. 2, 6, 7

Marco Lippi, Matteo Bertini, and Paolo Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882, 2013. 2

Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4): 276–281, 2004. 2

Selvaraj Vasantha Kumar. Traffic flow prediction using kalman filtering technique. *Procedia Engineering*, 187: 582–587, 2017. 2

Luis Leon Ojeda, Alain Y Kibangou, and Carlos Canudas De Wit. Adaptive kalman filtering for multi-step ahead traffic flow prediction. In *2013 American Control Conference*, pages 4724–4729. IEEE, 2013. 2

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 2, 7

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019. 2, 7

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 3

Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13668–13677, 2021. 3

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017. 6, 7

Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. Lsgcn: Long short-term traffic prediction with graph convolutional networks. In *IJCAI*, pages 2355–2361, 2020. 7, 8