DSTAN: Attention-Enhanced Dynamic Spatial-Temporal Network for Traffic Forecasting

Xunlian Luo[1], Chunjiang Zhu[2], Detian Zhang[1*], Qing Li[3]

[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Su Zhou, China.
[2]Department of Computer Science, UNC Greensboro, NC, USA.
[3]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.

*Corresponding author(s). E-mail(s): detian@suda.edu.cn;
Contributing authors: xlluo@stu.suda.edu.cn; chunjiang.zhu@uncg.edu; qing-prof.li@polyu.edu.hk;

Abstract

Traffic forecasting is an enduring research topic in the design of intelligent transportation systems and spatial-temporal data mining. Accurate prediction can help facilitate urban resource optimization and improve road efficiency. However, the complex spatial-temporal dependencies and dynamic urban conditions make it extremely challenging. Although many spatial-temporal modeling approaches have been proposed recently, they still suffer from the following three problems: (1) Inadequate modeling of temporal correlations; (2) Ignoring the fundamental fact that the location dependence of road networks changes dynamically over time; (3) Difficulty in extracting deeper spatial-temporal features layer by layer. In this paper, we propose a novel **D**ynamic **S**patial-**T**emporal **A**ttention-enhanced **N**etwork called DSTAN for traffic prediction. In DSTAN, we combine gated temporal units with trend-aware multi-head temporal attention to jointly capture local and long-range temporal dependencies. We also employ learnable node embeddings to extract heterogeneous information and integrate this with the spatial attention module to learn dynamic spatial correlations without any expert knowledge. Structurally, we stack multiple spatial-temporal blocks to improve the model's capability to identify complex patterns. Extensive experiments have been conducted on four widely used datasets, demonstrating that our method surpasses all baseline methods while exhibiting strong interpretability.

Keywords: traffic forecasting, urban computing, spatial-temporal time series, attention mechanism

1 Introduction

With the continuous growth of the urban population and the number of vehicles, the traffic network, serving as the arteries of the city, is under immense pressure. The high integration of technologies such as intelligent IoT, big data, and deep learning offers a promising solution for the development of Intelligent Transportation Systems (ITS) aimed at alleviating traffic congestion and optimizing resource allocation [1]. As a core technology of ITS, traffic prediction is crucial for maintaining the safety, stability, and reliability of the urban traffic environment [2, 3]. It aims to accurately forecast future traffic data, including traffic flow and speed, by leveraging historical traffic data collected from road network sensors [4]. Early works relied on the collection of univariate time series data from individual intersections, applying linear autoregressive statistical methods for analysis. With the extensive research on deep learning technologies, researchers have increasingly adopted Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and attention mechanisms to learn temporal correlations, while utilizing CNNs and Graph Convolutional Networks (GCNs) to capture spatial correlations in grid-based and graph-based traffic data [5, 6], respectively. Despite its success, traffic forecasting remains a challenging task due to it involves complex and dynamic spatial and temporal dependencies.

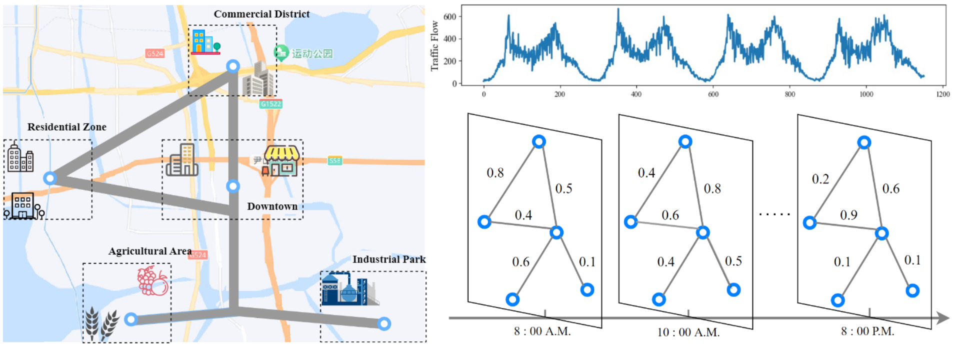


Fig. 1: An illustrating example for spatial-temporal correlations.

The traffic network is a complex system characterized by intricate connections and high spatial-temporal correlation, as depicted in Fig. 1. Temporal fluctuations in traffic data demonstrate persistence at a local scale. Over extended periods, traffic data exhibit similar changes to corresponding times on adjacent several days, indicating significant periodicity. This suggests that the temporal structure encompasses both local trends and global correlations. Spatially, the distribution of nodes within urban road networks is situated across various functional areas, exhibiting distinct data patterns due to spatial heterogeneity, and these nodes are interlinked and influence each other. Additionally, the right subfigure presents an example where spatial dependence between nodes is not constant but varies over time. For instance, the connection strength (traffic flow) between residential areas and commercial zones is significantly higher in the morning than in the evening, while the connection strength to the downtown is greater at night than during the day. This means that spatial node connections exhibit dynamic evolutionary characteristics. Therefore, effectively modeling these features is critical to achieve more accurate predictions.

Spatial-temporal graph neural networks (STGNNs) have made significant progress in traffic prediction by integrating sequence models with graph convolutional networks, effectively capturing both temporal and spatial correlations [7, 8]. CNN-based methods include STGCN [9], GraphWaveNet [10] and MTGNN [11], which typically employ exponential causal convolutions to eliminate the temporal dimension and utilize sequential or synchronous graph convolutions to encode knowledge into representations. RNN-based methods, such as DCRNN [12], AGCRN [13] and DDGCRN [14], replace the linear layers of recurrent neural networks with graph convolutions, treating the final hidden representation as a summary from the encoder. They then employ feedforward networks or decoders to generate predictions. However, the time steps of the training data are significantly lower than the number of nodes, resulting in a low density of temporal information due to limited samples. Using only a single sequence model further weakens the modeling process, leading to insufficient temporal modeling. In spatial modeling, the utilization of graphs has evolved from predefined structures to adaptive learning. There is still a scarcity of models that offer effective solutions for dynamic graphs that evolve over time, accompanied by robust interpretability. Attention-based methods, such as ASTGNN [2], STTN [15], DSTAGNN [16] and ST-WA [17], can learn dynamic spatial knowledge resembling complete graphs. Nonetheless, they often involve complex parameterization, making them susceptible to noise and overfitting. Additionally, recent representation learning methods such as STID [18] and ST-MLP [19] effectively identify heterogeneity through input embeddings. However, their overly simplistic downstream processing backbones fail to fully leverage the representational capabilities.

To solve the aforementioned shortcomings, in this paper, we present a **D**ynamic **S**patial-**T**emporal **A**ttention-enhanced **N**etwork (**DSTAN**) for traffic prediction. Specifically, we combine trend-aware temporal attention with gated temporal units to strengthen the model's ability to learn local trends and global dependencies in the temporal dimension. Recognizing that nodes have implicit heterogeneous information due to functional area distribution, we adopt learnable embeddings that undergo a dynamic clustering process during training to generate heterogeneous prototypes. We then integrate node embeddings with current input features and time of day to infer the dynamically evolving graph structure at each step, thereby learning dynamic spatial knowledge. Finally, these components are organically combined to construct spatial-temporal blocks, which help learn rich spatial-temporal representations layer by layer through stacking multiple layers. In summary, the main contributions of our work are as follows:

- Methodologically, we propose a novel framework called DSTAN that utilizes cascaded spatial-temporal blocks to construct the model backbone for traffic prediction.
- We integrate gated temporal units with a trend-aware temporal attention module to enhance the learning of temporal representations. This attention mechanism, informed by trend structural features, facilitates a more precise focus on variations in traffic dynamics.

- We capture the inherent heterogeneity of nodes through learnable node embeddings and integrate current input features to generate dynamic spatial relationships, thereby enhancing the model's adaptability and generalization in modeling spatial dependencies.
- Empirically, we conduct extensive experiments on four real-world traffic datasets and the results demonstrate that our method outperforms all baselines. Visualization experiments further illustrate its superior performance and strong interpretability.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature, while Section 3 delineates the problem formulation. In Section 4, we provide a detailed exposition of our proposed model. Subsequently, Section 5 presents comprehensive experimental results, and we conclude our discussion in Section 6.

2 Related Work

2.1 Traffic Prediction

Early traffic prediction is conducted as a univariate time series task. Statistical methods, including Historical Average (HA) and Autoregressive Integrated Moving Average (ARIMA) [20], utilized differencing and autoregression to forecast future traffic states based on the assumption of stationarity of the data. Some machine learning methods, like Support Vector Regression (SVR) [21] and XGBoost [22], use sliding window sampling on time series to convert the task into autoregressive supervised learning, capturing nonlinear representations. However, these methods only focus on temporal correlations and lack modeling of global road network features, often resulting in significant prediction errors.

With the impressive advancements of deep learning technologies in natural language processing and computer vision, numerous studies have attempted to apply 1D dilated convolutions [23], gated recurrent networks [24], and attention-based Transformers [25] to capture the temporal dynamics of traffic. Meanwhile, graphs, as a universal structure for representing real-world entities and their relationships, have been used in graph convolutional networks for non-Euclidean space modeling, achieving state-of-the-art performance in drug discovery and molecular prediction [26, 27]. Inspired by these advancements, numerous studies [14, 28] have begun to integrate graph convolutional networks with sequential models to jointly capture the temporal and spatial dependencies in traffic scenarios. Given the notable performance of graph convolutional networks, traffic prediction in terms of modeling spatial correlations can be divided into two folds [7]: the one involves designing a robust graph structure, which determines the learning quality of the graph convolutional layers; the other focuses on developing flexible and efficient graph convolution operators that learn effective spatial representations while avoiding over-smoothing.

In recent years, the application of Spatial-Temporal Graph Neural Networks (STGNNs) for modeling spatial-temporal data has increasingly become a prominent trend. STGCN [9] first proposes the sequential integration of spectral graph convolution and gated linear units to learn spatial and temporal dependencies in traffic data. DCRNN [12] combines diffusion convolution with GRU to model the spatial-temporal relationships in an encoder-decoder manner. AGCRN [13] and MTGNN [11] introduce the utilization of learnable embeddings to construct adaptive graph structures, effectively capturing the hidden spatial relationships. However, upon the completion of training, the graph structure remains unchanged since the parameters are not changed during the testing phase. STGODE [29] and STG-NCDE [30] employ ordinary or neural-controlled differential equations to learn continuous spatial-temporal features. PDFormer [31] adopts masked spatial delay-aware attention and temporal attention modules to capture the spatial-temporal features inherent in traffic data. Nevertheless, they fail to adequately capture the spatial correlations that vary over time.

2.2 Attention Mechanism

The attention mechanism draws on the visual selective attention mechanism in human evolution [25]. Its core goal is to identify information that is more crucial to the target task and more valuable for knowledge reasoning from a large amount of information. This makes it superior to convolutional and recurrent networks in tasks of massive data such as Natural Language Processing [32] and Computer Vision [33]. In the realm of spatial-temporal data mining, models such as DSANet [34], ASTGNN [2], and STAEformer [35] have demonstrated the efficacy of the attention mechanism in the dynamic modeling of traffic patterns.

The self-attention mechanism is a fundamental approach that empowers models to establish self-referential relationships and make adjustments within a single input. The query, key, and value matrices are all extracted from the same sequence of symbolic representations. By computing similarity scores between query and key matrix token positions, the token representation at each position can interact with all other positions to capture global correlations and dependencies [25, 36]. Transformer relies exclusively on a model of the self-attention mechanism and has achieved state-of-the-art performance across a wide range of sequential tasks, including text generation [37] and speech processing [38]. It transforms the representation of symbols in a sequence of sequential inputs by scaled dot-product to compute a series of contextual information vectors. Since the representation of each symbol is directly affected by the representations of all other symbols, this produces an effective global receptive field, solving the problem that RNNs cannot be easily parallelized and CNNs cannot efficiently capture long-term dependencies [2]. In traffic prediction, the self-attention mechanism offers a highly flexible framework by effectively processing different dimensions of spatial-temporal data [15, 39, 40]. It not only captures long-range temporal correlations in traffic patterns but also facilitates the modeling of dynamic spatial dependencies. However, complex parameter optimization often causes the model to over-fit the noise, resulting in poor accuracy.

3 Preliminaries

Definition 1: The traffic sensors deployed on the road network are formulated as the graph $\mathcal{G} = (V, E, A)$, where V is the set of $|V| = N$ nodes, E is the set of node-connected edges and $A \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix that quantifies the

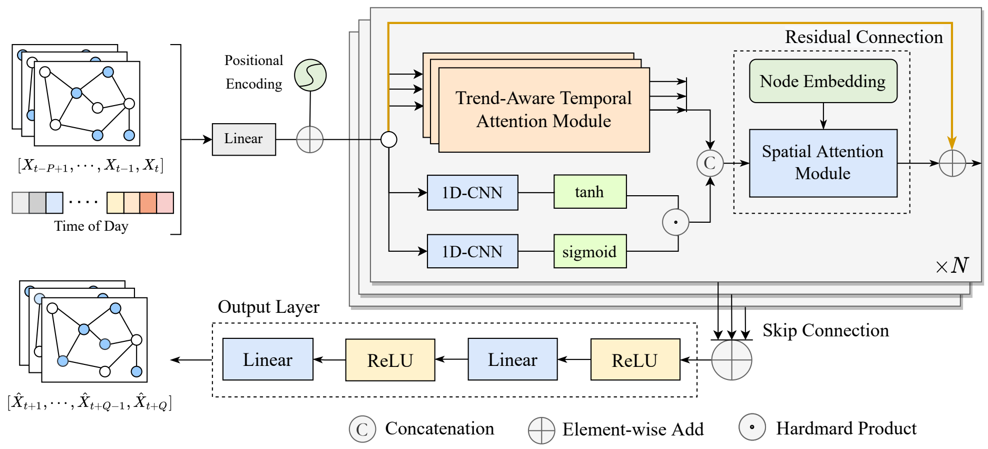


Fig. 2: The framework of the proposed DSTAN.

proximities between nodes, including geographical distances, semantic relationships, or adaptive embedding similarity measures between pairs of nodes.

Definition 2: The graph signal (or feature matrix) for all nodes in the traffic network at time step t can be denoted as the 2D tensor $X^{(t)} \in \mathbb{R}^{N \times D}$, where D is the feature dimension of each node (e.g., flow, speed, occupancy). Similarly, we denote $X^{(t:t+P)} \in \mathbb{R}^{P \times N \times D}$ as the graph signal of all nodes over P consecutive time intervals.

Problem Statement: Traffic forecasting can be summarized as a multivariate time series forecasting task under auxiliary graph knowledge $\mathcal{G}$. Its objective is to establish a mapping $\mathcal{F}_\Theta(\cdot)$ from historical observations to future data, defined as follows:

$$\left[X^{(t-P+1)}, \cdots, X^{(t-1)}, X^{(t)}; \mathcal{G}\right] \xrightarrow{\mathcal{F}_\Theta(\cdot)} \left[\widehat{X}^{(t+1)}, \widehat{X}^{(t+2)}, \cdots, \widehat{X}^{(t+Q)}\right] \tag{1}$$

where P represents the length of the historical sequence and Q is the length of the predicted sequence.

4 Proposed Methodology

In this section, we first provide an overview of the framework for our proposed method, followed by a detailed introduction of each component's technical specifics. Finally, we conclude with a description of the training procedure with loss optimization.

4.1 Framework of DSTAN

Fig. 2 illustrates the framework of DSTAN, which consists of several stacked spatial-temporal blocks and an output layer. The input features are first appended with temporal information (e.g. Time of Day) and positional encoding of the time axis, then transformed by a linear layer to raise the channel dimension, and finally mapped to the output space via multiple cascaded spatial-temporal layers. Temporally, we adopt gated temporal units and trend-aware multi-head temporal attention to handle local and long-range temporal dependencies separately. Spatially, the node-adaptive parameter embeddings learn the implicit heterogeneous information of nodes in an intrinsic, data-driven manner, eliminating the need for prior domain-specific knowledge to describe it. These embeddings are integrated with the spatial attention module to capture dynamic spatial dependencies, facilitating the model's adaptability and generality. Additionally, motivated by the structure of WaveNet [41], we employ residual connections and skip connections to expand the depth of the model and layered architecture to enhance its nonlinear representation of complex spatial-temporal patterns.

4.2 Gated Temporal Unit

Future traffic states are closely related to adjacent historical observation data, exhibiting pronounced local temporal trends. Previous research has primarily utilized recurrent neural networks (RNNs) to capture these sequential relationships. However, RNNs encounter challenges such as time-consuming iterations, slow adaptation to dynamic changes, and issues with gradient instability [9]. In contrast, convolutional neural networks (CNNs) leverage one-dimensional convolutional kernels to extract local temporal receptive fields, offering significant advantages. CNNs are adept at effectively identifying local trend variations within time series, while parameter sharing enhances the ability to generalize to patterns across different time steps during the learning process. Moreover, gating mechanisms have also proven effective in controlling information flow within temporal convolution layers [42]. Therefore, we first utilize a set of fixed-size gated convolution units along the time axis to capture the local temporal behavior of traffic. Mathematically, given a 1D input sequence $x \in \mathbb{R}^T$ and a filter $f \in \mathbb{R}^K$, the one-dimensional convolutional layer as below:

$$x \star f = \sum_{k=0}^{K-1} x(T-k)f(k) \tag{2}$$

Based on this, given the input feature matrix $X \in \mathbb{R}^{C\times N\times T}$ for all nodes in graph $\mathcal{G}$, the temporal convolution layer explores the features of K_t adjacent time steps. To ensure a consistent input length, we apply $(K_t - 1)$ zero padding to the left side of the sequence. The equation for the Gated Temporal Unit is defined as follows:

$$H_g = g(\Theta_1 \star X_{:,:,t} + b_1) \odot \sigma(\Theta_2 \star X_{:,:,t} + b_2) \tag{3}$$

where Θ_1, Θ_2, b_1 and b_2 are all convolutional kernel parameters, $\odot$ is the element-wise product. $g(\cdot)$ and $\sigma(\cdot)$ are the tanh and sigmoid activation functions used to control the ratio of sequence information that passes through, respectively. This gated convolution unit enhances the model's sensitivity to local temporal dependencies.

4.3 Trend-Aware Temporal Attention Module

In the raw self-attention mechanism, the temporal dependence of time series is often aggregated via point-wise products, which makes it difficult to find realistic temporal trend features. As presented in Fig. 3, the left one illustrates how the traditional self-attention mechanism may erroneously match points A and B with similar values while

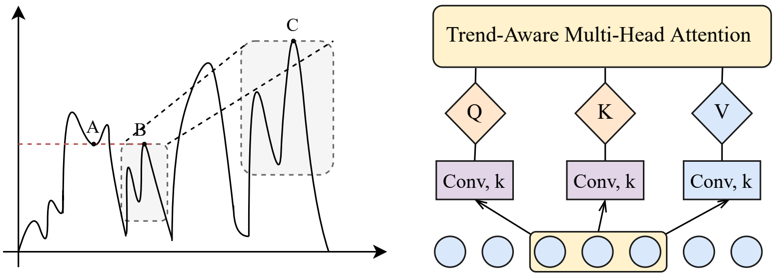


Fig. 3: Trend-Aware Temporal Attention Module.

neglecting the trend information present in their surrounding context. This oversight impedes the effective extraction of long-range temporal features [43]. In contrast, the right one introduces local trend awareness into the self-attention mechanism, and enables a more accurate alignment of data points B and C, as they exhibit highly similar data trends over a specific period.

In light of this, we still employ one-dimensional convolution to simulate the trend characteristics of the time series structure, thereby optimizing the original attention mechanism, which enables the model to better comprehend the dynamic patterns and long-range dependencies of traffic data. Specifically, we first incorporate positional encoding into the original input to assist the attention module in marking the relative positional relationships of the sequence. Given input feature matrix $H = X + E_{TP}$, where E_{TP} is defined as in Equation (4):

$$E_{TP} = \begin{cases} \sin(t/10000^{2i/d_{model}}), & \text{if } \ t = 0, 2, 4, \ldots \\ \cos(t/10000^{2i/d_{model}}), & \text{otherwise} \end{cases} \tag{4}$$

Then we adopt a 1D convolutional kernel to smooth the input sequence to obtain the Q, K, and V matrices with trend information referred to Equation (2). To further quantify the contextual relationships between sequence elements, we compute the similarity between the Query and Key matrices. Consequently, the Trend-Aware Temporal Attention Module can be represented as follows:

$$\text{TrendAttention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{5}$$

where d_k serves as a scaling factor, and the softmax function is used to normalize the weight scores.

In practice, we typically utilize multiple attention heads to focus on inputs from different representation subspaces, thereby obtaining more stable and enriched latent information. Finally, the representations from the different heads are concatenated and subsequently projected to produce the final result, calculated as follows:

$$\begin{aligned} &\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \cdots, \text{head}_h)W^O, \\ &\text{where } \ \text{head}_i = \text{TrendAttention}(\Phi_i^Q \star H, \Phi_i^K \star H, \Phi_i^V \star H). \end{aligned} \tag{6}$$

where $\star$ indicates the convolution operator and Φ_i^Q, Φ_i^K and Φ_i^V are the parameters of convolution kernels referenced to Eq. 2. $W^O \in R^{D \times d}$ is projection matrix for the linear transformation, $D = h \times d_k$, h is the number of heads, and d_k is the dimension of the hidden representation.

Through the trend-aware temporal attention module, we can more effectively capture the temporal dynamics and global dependencies of traffic state. Subsequently, we concatenate the outputs H_g and H_t from the **Gated Temporal Unit** and the **Trend-Aware Attention Module** along the channel dimension, merging the temporal features from both components. Then perform a linear transformation to obtain the hidden features as below:

$$H_T = \text{Conv}_{1\times 1}(H_g || H_t) \tag{7}$$

The processed features H_t are utilized both as outputs for the model's skip connections and as inputs for the dynamic spatial attention module.

4.4 Spatial Attention Module

The traffic conditions of urban road networks are highly dynamic and time-varying. Traditional GCNs cannot capture such patterns through signal propagation on fixed graphs [44, 45]. The adaptive graph is constructed based on training data, which mitigates biases introduced by prior knowledge to some extent, it still overlooks the dynamic changes in edge weights associated with the connectivity of road network nodes over time. Although the spatial attention mechanism can dynamically compute weights based on real-time input features, redundant and short-sequence feature information often fails to accurately capture spatial dynamics.

To better capture the dynamic evolution characteristics of urban spatial correlation, we propose a spatial attention module that integrates joint node embeddings. This module encodes static node heterogeneous features through a set of learnable node embeddings, where the learning of these embeddings effectively performs a dynamic clustering process, resulting in the organic formation of clusters among nodes with similar heterogeneous features. Considering that the spatial associations of nodes are also closely related to time state and real-time data, we combine node embeddings, time information (Time of Day), and real-time input features to infer node connections at each time step. The detailed structure of this module is shown in Fig. 4.

Mathematically, we first represent the spatial graph for the temporal domain T, given the learnable embeddings of the nodes $E_v \in \mathbb{R}^{N \times d}$, where d denotes the dimensionality of the node embeddings. Subsequently, we concatenate the broadcasted embedding vectors with the input features $H_T \in \mathbb{R}^{T \times N \times d}$ to form the basis vector for inferring the dynamic spatial graph. Here, we utilize a scaled dot-product similar to an attention mechanism to compute the dynamic adjacency matrix $\mathbb{E}(t)$ at each time step t, expressed as follows:

$$\mathbb{E}(t) = \frac{\langle f_{s,1}(E_v || H_{v,t}), f_{s,2}(E_v || H_{v,t}) \rangle}{\sqrt{d}} \tag{8}$$

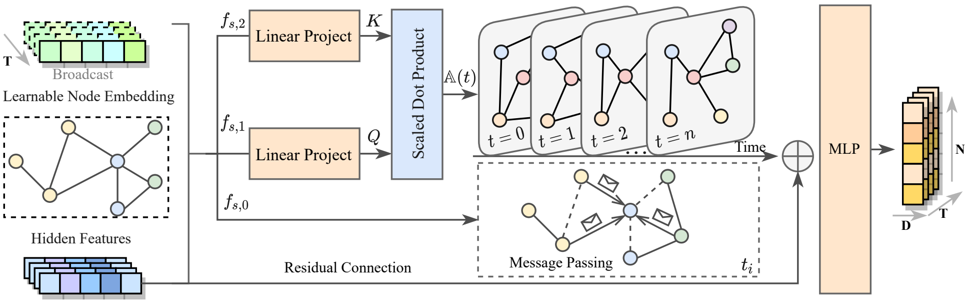


Fig. 4: Spatial Attention Module.

where || denotes the concatenation operation along the channel dimension, $\langle\cdot,\cdot\rangle$ represents the dot product operation, $f_{s,1}$ and $f_{s,2}$ are learnable linear projection layers, and d refers to the dimensionality of the input channels.

Given the time step $t \in \{1, 2, \cdots, T\}$, $e_{vi,vj} \subset E(t)$ indicates the dynamic weight of node v_i and v_j, we further normalize the weight scores through softmax function and recalculate the distribution. The formula is as below:

$$\mathbb{A}(t) = \text{softmax}(e_{v_i,v_j}) = \frac{\exp(e_{v_i,v_j})}{\sum_{v_k \in V} \exp(e_{v_i,v_k})} \tag{9}$$

where $\mathbb{A}(t)$ is the dynamic adjacency matrix generated by combining learnable node embeddings with real-time features, which dynamically assign weights among different nodes (i.e., sensors) at various periods.

Similar to graph convolution operators, we utilize matrix multiplication between the dynamic graph $\mathbb{A}(t)$ and the original input features H_T to aggregate information from neighboring nodes, computed as follows:

$$H_S = f_{s,4}\left(f_{s,3}\left(\text{GCN}(\mathbb{A}(t), H_T) \mid\mid H_T\right)\right) \tag{10}$$

where $f_{s,3}$ represents the feature linear transformation layer, while $f_{s,4}$ denotes the Dropout function. Additionally, we retain a proportion of the original features in the output of the module to mitigate the issue of over-smoothing in graph convolution.

Overall, this module not only takes into account the static implicit spatial features of the nodes but also incorporates real-time feature inputs and time encoding into the construction of the dynamic graph. This integration allows for more comprehensive modeling of the nodes' implicit heterogeneous characteristics and spatial dynamics, thereby mitigating the model's reliance on prior knowledge graphs.

4.5 Loss Optimization Process

The spatial-temporal prediction task aims to learn the nonlinear function $\mathcal{F}_\Theta$ based on the gradient descent of the loss. The objective of optimizing the $\mathcal{L}$ is as follows:

$$\Theta^* = \arg\min_{\Theta} \ \nabla\mathcal{L}(\mathcal{F}(A, X_{t-P-1:t}; \Theta), X_{t+1:t+Q}) \tag{11}$$

where Θ represents the parameters to be optimized within the mapping function. In practice, we adopt the Mean Absolute Error (MAE) to evaluate the relative loss between the sample labels $\overline{X_i}$ and the model predictions $\widehat{X_i}$. The loss can be articulated as below:

$$\mathcal{L} = \frac{1}{\Omega} \sum_{i=1}^{\Omega} |\overline{X_i} - \widehat{X_i}| \tag{12}$$

where Ω represents the number of samples in the dataset.

Here we illustrate the computational workflow of each component in DSTAN and the detailed batch iteration optimization through Algorithm 1.

Algorithm 1 Training algorithm of DSTAN.

```
Input: The traffic dataset $O$, the initialized DSTAN model $\mathcal{F}_{\Theta}(\cdot)$ with $\Theta$, learning rate $\gamma$, batch size $b$, the number of spatial-temporal blocks $L$.
Output: learned model.
set $epoch = 1$
repeat
    for sample a batch ($X \in \mathbb{R}^{b \times P \times N \times D}$, $\overline{X} \in \mathbb{R}^{b \times Q \times N \times D}$ ) from $O$ do
        map features to a high-dimensional space $X = X \cdot W + b$.
        add positional encoding $H^{(1)} = X + E_{TP}$.
        for $i$= 1 to $L$ do
            take $H^{(i)}$ into the Gated Temporal Unit yields $H_g^{(i)}$.
            take $H^{(i)}$ into the Trend-Aware Temporal Attention Moduleyields $H_t^{(i)}$.
            aggregate temporal features $H_T^{(i)} = \text{Dropout}(\text{MLP}(H_g^{(i)} || H_t^{(i)}))$.
            skip connection outputs the features of the $i$-th block $H_T^{(i)}$.
            put $H_T^{(i)}$ into the dynamic Spatial Attention Module yields $H_S^{(i)}$.
            residual connection $H^{(i+1)} = H_S^{(i)} + H^{(i)}$.
        end for
        merge the output of skip connections $H_T = \text{Sum}(H_T^{(1)}, H_T^{(2)}, \cdots, H_T^{(L)})$.
        output layer produces $\widehat{X} = \text{MLP}(\text{ReLU}(\text{MLP}(H_T)))$.
        compute MAE Loss $\mathcal{L}(\widehat{X}, \overline{X})$.
        back propagation and update parameters $\Theta$ according to $\mathcal{L}$.
    end for
    $epoch = epoch + 1$.
until convergence
```

5 Experiments

In this section, we first introduce the experimental setup including the dataset used, the baseline methods and the evaluation metrics. Then the experimental results of the method are presented and analyzed. Finally, we investigate the effects of model hyperparameters, module ablation, and empirical analysis of graph learning.

5.1 Experimental Settings

Datasets. To evaluate the effectiveness of our proposed method, we utilize four publicly available traffic speed datasets published by DCRNN [12] and STGCN [9]. These datasets include METR-LA, with four months of data from 207 highway sensors in Los Angeles County; PEMS-BAY, containing six months of data from 325 sensors in the Bay Area; PEMSD7(M) and PEMSD7(L) containing 44 days of data collected from 225 and 1026 sensors on the California Transportation Performance Management System [46]. They are all also aggregated into 5-minute intervals from 30-second data samples. The statistics of the datasets are presented in Table 1.

Table 1: Summary of traffic speed datasets.

Dataset	Nodes	Edges	Samples	Time Range	MissingRatio
METR-LA	207	1515	34,272	03/2012 - 06/2012	8.109%
PEMS-BAY	325	2369	52,116	01/2017 - 05/2017	0.003%
PEMSD7(M)	228	1132	12,672	05/2012 - 06/2012	0.000%
PEMSD7(L)	1026	1132	12,672	05/2012 - 06/2012	0.000%

In our experiment, we apply Z-Score Normalization [5] to scale the raw inputs of the samples to accelerate convergence. Following previous work, we divide the METR-LA and PEMS-BAY datasets into training, validation, and test sets at a ratio of 7:1:2. For PEMSD7(M) and PEMSD7(L) datasets, we use the first 60% of the data for training, 20% for validation, and the last 20% for testing. We aim to predict the data for the future hour based on the observations from the previous hour, i.e., P and Q are both set to 12.

Baseline Methods. We compare our DSTAN with several widely adopted baselines, including statistical-based models HA and ARIMA [20], as well as classic and latest STGNN-based models in this field. Some of the baselines are introduced as follows:

- DCRNN (2017 ICLR) [12]: It replaces the linear network of gated recurrent units with bidirectional diffusion graph convolutions and combines them in an encoder-decoder architecture.
- GWNET (2019 IJCAI) [10]: It integrates diffusion graph convolutions with gated 1D dilated convolutions and proposes a self-adaptive adjacency matrix.
- MTGNN (2020 KDD) [11]: It adopts a self-learned graph structure, mix-hop propagation layers and dilated inception layers to capture the spatial-temporal correlations among multiple time series.
- AGCRN (2020 NeurIPS) [13]: It employs a graph learner with specialized node adaptive parameters and integrates graph convolution with GRUs.
- STG-NCDE (2022 AAAI) [30]: It introduces two NCDEs tailored for temporal and spatial features processing, seamlessly integrating them into a unified framework.
- STID (2022 CIKM) [18]: It combines various spatial-temporal embeddings with MLPs to propose a minimalist prediction framework.

- ST-WA (2022 ICDE) [17]: It proposes a spatial-temporal aware method that jointly learns specific location and time-varying model parameters from encoded stochastic variables.
- DDGCRN (2023 PR) [14]: It utilizes time-varying features and temporal embeddings to generate dynamic graphs and designs a signal decomposition network in conjunction with RNNs.
- Trafformer (2023 AAAI) [47]: It unifies the modeling of spatial and temporal information into a transformer-like model and generates multi-step predictions with a generative decoder.
- PDFormer (2023 AAAI) [31]: It adopts masked spatial delay-aware attention and temporal attention modules to jointly model complex spatial-temporal correlations.
- PGCN (2024 TITS) [48] It constructs a progressive adjacency matrix by learning the trend similarity between graph nodes and combines it with dilated causal convolution for spatial-temporal prediction.

Parameters Setup. Across all datasets, we set the dimension of hidden features to 32. The convolution kernel of the gated temporal unit is set to 3, and the head of the trend-aware temporal attention module is set to 4. The optimal number of spatial-temporal blocks is searched in $\{2, 3, 4, 5\}$. The dropout ratio in the spatial attention module is specified as 0.3. Additionally, we conduct a series of hyperparameter tuning experiments to identify the optimal node embedding size and trend convolution kernel for DSTAN across various datasets. We use the Adam optimizer with an initial learning rate of 0.001 and the batch size of the data is 64. The maximum training epochs is set to 200, with an early stop patience set to 20. All following experiments are conducted on NVIDIA Tesla V100 GPUs.

Throughout the experiments, we use three commonly used evaluation metrics to evaluate the performance of our method, i.e. Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) [10]. Due to the presence of noisy values in the samples, we adopt a zero-masking strategy to avoid obtaining abnormal MAPE when calculating these metrics. Smaller values of these indicate better performance of the method.

5.2 Experimental Results

Table 2 presents a comparative analysis of the prediction performance of DSTAN and baseline methods on the METR-LA and PEMS-BAY for 15, 30, and 60 minutes ahead (Horizon 3, 6, 12). Table 3 illustrates the average performance of all models over 12 predicted timesteps on the PEMSD7(M) and PEMSD7(L) datasets. From the results, we can conclude the following observations: (1) DSTAN outperforms statistical models by a large margin such as HA and ARIMA, which neglect the spatial characteristics of traffic data and rely on linear autocorrelation, making it challenging to identify complex patterns. (2) As early GNN-based models, STGCN and DCRNN exhibit significant performance improvements over models that consider only temporal relationships. However, pre-defined graph structures hinder their ability to identify complex spatial relationships. Recent advanced methods, including GWNET, MTGNN, and AGCRN, utilize adaptive graph learning to mitigate pre-set

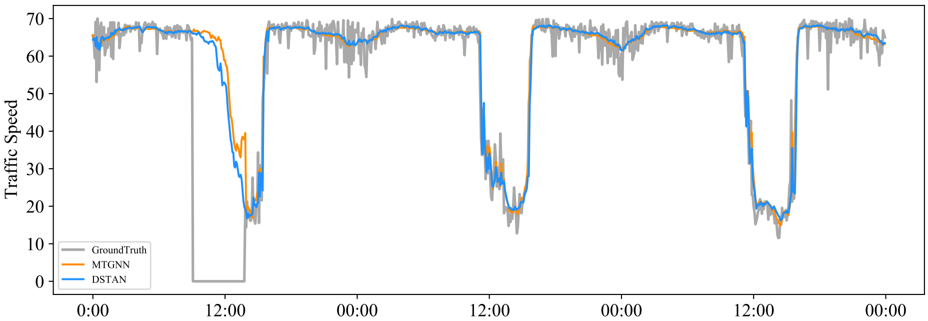
Table 2: Performance comparison with baseline models on METR-LA and PEMS-BAY.

Dataset	**Method**	**Horizon 3**			**Horizon 6**			**Horizon 12**		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	HA	4.16	7.80	13.00%	4.16	7.80	13.00%	4.16	7.80	13.00%
	ARIMA	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	FC-LSTM	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	STGCN	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	DCRNN	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	GTS	2.75	5.27	7.12%	3.14	6.33	8.62%	3.59	7.44	10.25%
	GWNET	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
	MTGNN	2.69	5.18	6.88%	3.05	6.17	8.19%	3.49	7.23	9.87%
	STID	2.82	5.53	7.75%	3.19	6.57	9.39%	3.55	7.55	10.95%
	ST-WA	2.89	5.62	7.66%	3.25	6.61	9.22%	3.68	7.59	10.78%
	Trafformer	2.78	5.35	7.32%	3.05	6.18	8.67%	**3.41**	7.17	9.96%
	PDFormer	2.83	5.45	7.77%	3.20	6.46	9.19%	3.62	7.47	10.01%
	PGCN	2.70	5.16	6.98%	3.08	6.22	8.38%	3.54	7.36	9.94%
	DSTAN	**2.68**	**5.12**	**6.87%**	**3.04**	**6.07**	**8.09%**	3.46	**7.13**	**9.62%**
PEMS-BAY	HA	2.88	5.59	6.80%	2.88	5.59	6.80%	2.88	5.59	6.80%
	ARIMA	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM	2.05	4.19	4.08%	2.20	4.55	5.20%	2.37	4.96	5.70%
	STGCN	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	DCRNN	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	GTS	1.37	2.92	2.85%	1.72	3.86	3.88%	2.06	4.60	4.88%
	GWNET	**1.30**	**2.74**	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	MTGNN	1.32	2.79	2.77%	1.65	3.74	3.69%	1.94	4.49	4.53%
	STID	1.31	2.79	2.78%	1.64	3.73	3.73%	1.91	4.42	4.55%
	ST-WA	1.37	2.88	2.86%	1.70	3.81	3.81%	2.00	4.52	4.63%
	Trafformer	1.31	2.83	2.92%	1.61	3.74	3.82%	**1.88**	4.38	4.59%
	PDFormer	1.32	2.83	2.78%	1.64	3.79	3.71%	1.91	4.43	4.51%
	PGCN	1.30	2.73	2.72%	1.62	**3.67**	3.63%	1.92	4.45	4.45%
	DSTAN	1.31	2.84	**2.71%**	**1.61**	3.68	**3.57%**	1.89	**4.34**	**4.35%**

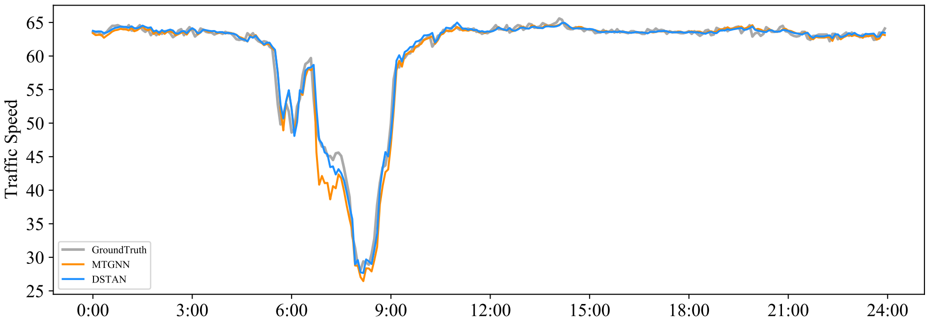
Table 3: Performance comparison with baseline models on PEMSD7(M/L).

Dataset	Metric	STGCN	DCRNN	GWNET	AGCRN	STG-NCDE	PDFormer	DDGCRN	DSTAN
PEMS-M	**MAE**	3.86	3.83	3.19	2.79	2.68	2.81	2.59	**2.57**
	RMSE	6.79	7.18	6.24	5.54	5.39	5.60	5.21	**5.08**
	MAPE	10.06%	9.81%	8.02%	7.02%	6.76%	7.06%	6.48%	**6.43%**
PEMS-L	**MAE**	3.89	4.33	3.75	2.99	2.87	2.92	2.79	**2.78**
	RMSE	6.83	8.33	7.09	5.92	5.76	5.90	5.68	**5.50**
	MAPE	10.09%	11.41%	9.41%	7.59%	7.31%	7.54%	7.06%	**6.94%**

bias. Nonetheless, they still ignore the time-varying nature of spatial correlations. Our proposed method, DSTAN, not only accounts for temporal dependence but also models dynamic spatial dependencies, achieving superior performance across most metrics. (3) As baselines with the same type (i.e. attention mechanism mainly used in their backbone), ST-WA, Trafformer, and PDFormer are somewhat less effective than DSTAN in modeling spatial-temporal correlations. Although ST-WA and DDGCRN learn specific parameters for each location and time step from the current input, they either neglect or fail to effectively utilize the inherent implicit spatial features of the nodes. As a result, our DSTAN demonstrates better forecasting performance than others.



(a) Node 84 in METR-LA



(b) Node 75 in PEMS-BAY

Fig. 5: Visualization comparison of ground truth and model predictions for METR-LA and PEMS-BAY.

To further intuitively validate the performance of our proposed method, we conduct a visualization analysis. Fig. 5 visualizes the variations in speed predictions by the model compared to the ground truth for a specific node over several days on two datasets. In comparison to the gray realistic curve, it is evident that both models accurately predict the future trends in speed variations. Due to the complex distribution of the METR-LA data and the presence of many non-stationary missing values (which were filled with zeros during data preprocessing), our method is still able to learn stable patterns, effectively fitting the trends of the missing values and demonstrating robustness. Additionally, the data from PEMS-BAY indicates that the period from 6:00 to 9:00 likely represents peak traffic hours, during which speed trends significantly decrease. In this context, DSTAN aligns more closely with actual expectations than MTGNN, demonstrating superior predictive performance.

5.3 Parameter Sensitivity Analysis

Fig. 6 and 7 illustrate the effect of the two critical model hyperparameters on the prediction performance. Analysis of the error curves in Fig. 6 reveals that the optimal trend convolution kernel size for METR-LA, PEMS-BAY, PEMSD7(M), and PEMSD7(L) is all 5. An appropriately selected convolution kernel size strikes a balance between the temporal receptive field and the model's complexity, thereby facilitating

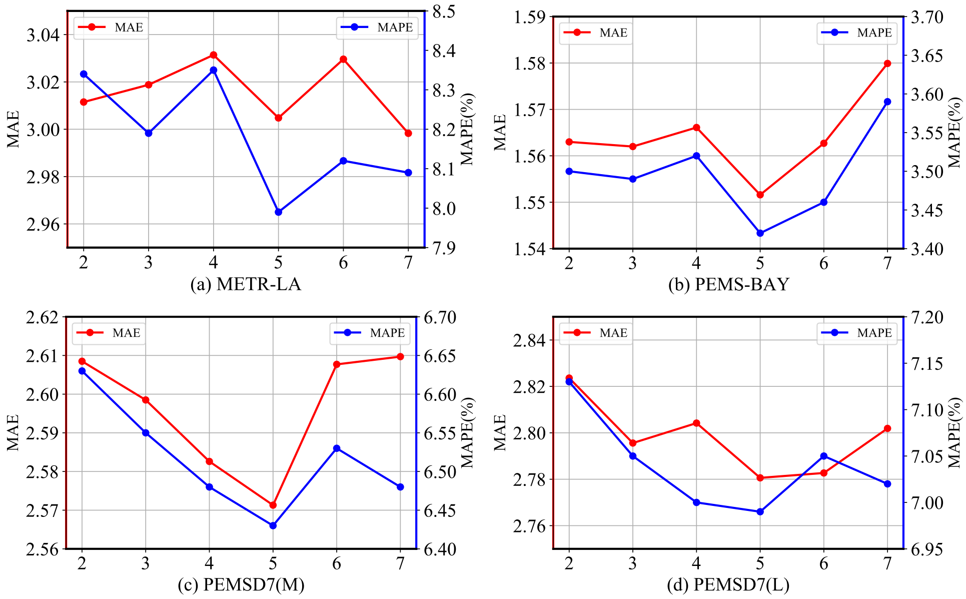


Fig. 6: Effect of different trend convolution kernels on performance.

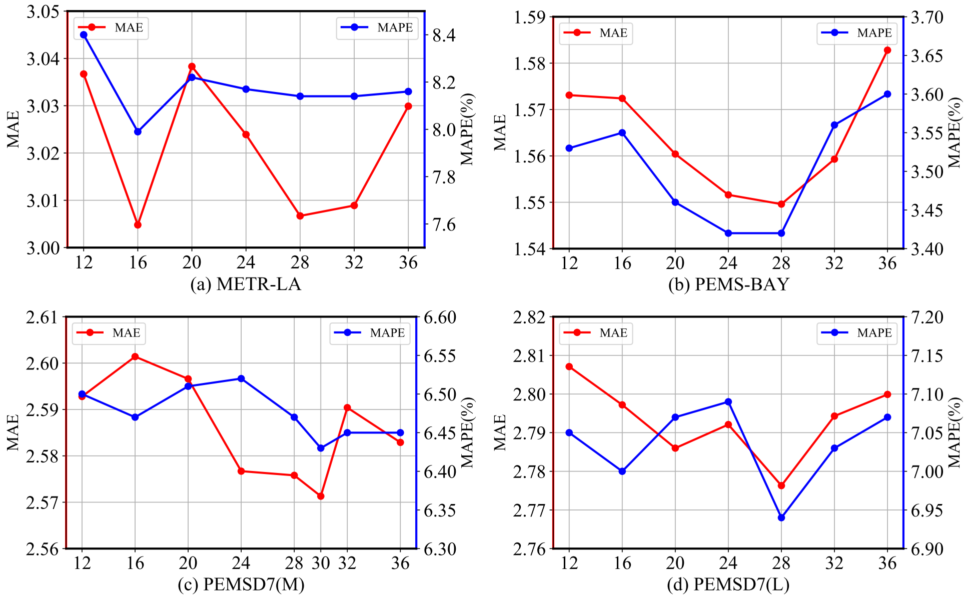


Fig. 7: Effect of different node embedding sizes on performance.

the effective capture of temporal trend similarities within the traffic data. Similarly, we can identify the optimal parameters for node embeddings in each dataset by analyzing the minimum error curves. The optimal values for METR-LA, PEMS-BAY, PEMSD7(M), and PEMSD7(L) are 16, 28, 30, and 28, respectively. The value is related to the density of heterogeneous information within the dataset. If it is too small, it may fail to capture the implicit features of the node domains adequately. Conversely, if it is too large, it may lead to overfitting, resulting in increased error.

5.4 Ablation Study

To gain deeper insights into how the various components impact the overall performance, we conduct an ablation study on the model variants on the METR-LA and PEMS-BAY datasets. The variants of DSTAN are referred to as follows:

- DSTAN w/o SAttn: It removes the spatial attention module associated with the spatial-temporal blocks, thereby eliminating the modeling of spatial correlations.
- DSTAN w/o NE: It replaces the node embeddings with a spatial positional encoding.
- DSTAN w/o TConv: It removes trend convolution operators from the trend-aware temporal attention module and then restores it to the original attention structure.
- DSTAN w/o GTU: It excludes the gated temporal units from DSTAN.

Table 4: Ablation study on each component.

Module	METR-LA			PEMS-BAY		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
DSTAN w/o SAttn	3.58	7.18	10.28%	1.81	4.07	4.11%
DSTAN w/o NE	3.06	6.14	8.44%	1.57	3.49	3.50%
DSTAN w/o TConv	3.05	6.08	8.31%	1.60	3.53	3.53%
DSTAN w/o GTU	3.04	6.04	8.18%	1.57	3.50	3.48%
DSTAN	**3.00**	**5.96**	**7.99%**	**1.55**	**3.49**	**3.42%**

Table 4 presents the performance differences of DSTAN and its variants on the METR-LA and PEMS-BAY. It is very obvious that when we remove the spatial attention module, the prediction performance of the model decreases dramatically. This indicates that modeling spatial dependencies is crucial for improving prediction accuracy. Replacing learnable node embeddings with static spatial encoding resulted in a slight decline in predictive accuracy, as static encoding relies on prior knowledge and cannot automatically adapt to the heterogeneity of nodes based on data characteristics. Similarly, the removal of TConv or GTU leads to increased prediction error. In contrast, jointly modeling local and global temporal features yields superior results compared to using them in isolation. Collectively, these design choices underscore the holistic and indivisible nature of the DSTAN, enabling its superior spatial-temporal forecasting performance.

5.5 Interpretability of Graph Learning

The spatial relationships in urban road networks are complex and diverse. We conduct different equations for graph learning to analyze their predictive performance. In the experiment, we substitute the spatial attention module in this paper with the

Table 5: Comparison of different graph learning equations on METR-LA. The adaptive graph computes similar proximity by node embedding E_1, E_2 or E.

Method	Equation	MAE	RMSE	MAPE
Pre-defined Graph	—	3.0980	6.12	8.57%
Adaptive Graph	$A = softmax(ReLU(E_1 E_2^T))$	3.0603	6.10	8.35%
Uni-directed Graph	$A = softmax(ReLU(E_1 E_2^T - E_2 E_1^T))$	3.0650	6.13	8.36%
Self-attention Graph	$A = softmax\left(\frac{EW_1(EW_2)^T}{\sqrt{d}}\right)$	3.0578	6.08	8.28%
Ours	$A = softmax\left(\frac{(E\|\|H)W_1((E\|\|H)W_2)^T}{\sqrt{d}}\right)$	**3.0048**	**5.96**	**7.99%**

bidirectional diffusion graph convolution operator utilized in DCRNN, and employ learnable parameters E (or E_1, E_2) to construct an adaptive adjacency matrix. We repeat the experiment five times to calculate the average performance of different graph learning methods on MAE, RMSE and MAPE, as shown in Table 5. In comparison to pre-defined graphs, adaptive graphs more effectively identify hidden spatial relationships that may be absent from prior knowledge, thereby yielding lower MAE, RMSE, and MAPE. Adaptive Graph, Uni-directed Graph, and Self-attention Graph perform similarly across the three metrics. During testing, they cannot adjust their graph structures based on current inputs, limiting their adaptability to dynamic traffic scenarios. In contrast to suboptimal methods, our dynamic graph learning within the spatial attention module demonstrates improvements of 1.7%, 2.0%, and 3.5% in MAE, RMSE, and MAPE, respectively.

To further validate the contribution of our method in learning dynamic spatial knowledge, we select a continuous sequence of numbered nodes from the test data and analyze their spatial dependency variations over a specific period. As illustrated in Fig. 8, we present the correlation heatmaps for node pairs at the current time, as well as at 30 minutes later, 60 minutes later, 90 minutes later, 2 hours later, and 3 hours later. Darker colors indicate a stronger correlation between nodes, while lighter colors signify a weaker correlation. From a macro perspective, the spatial connections between nodes exhibit a degree of stability and invariance. However, a closer examination at the micro level reveals that the connection strength between certain node pairs fluctuates over time, either strengthening or weakening. Without loss of generality, we take the target highlighted by the red box in the figure as an example, illustrating how its spatial correlation continuously adjusts across different time phases. Consequently, this phenomenon substantiates the necessity of dynamic graph learning within the spatial attention module, which is essential for adapting to spatial dependencies changing over time in real-world scenarios.

6 Conclusion

In this paper, we propose a novel spatial-temporal graph model called DSTAN. It integrates gated temporal units with trend-aware attention to effectively capture both local and global temporal dependencies. We also adopt adaptive node embeddings to learn implicit node-heterogeneous information, which is then combined with input features

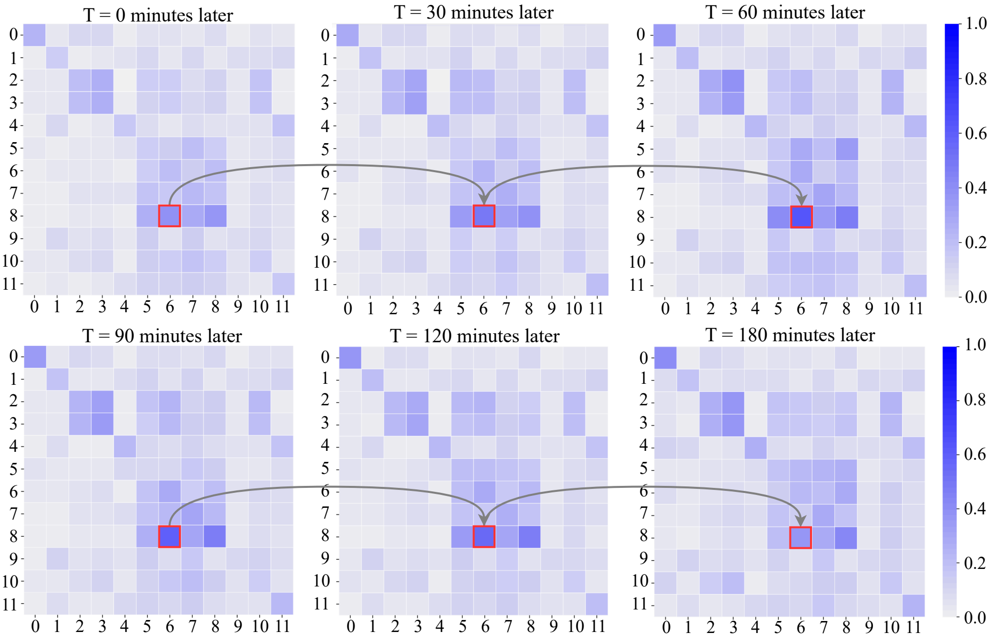


Fig. 8: Spatial correlation heatmap of different time periods on METR-LA.

in the spatial attention module to infer a graph structure that dynamically adjusts according to the data characteristics. This significantly enhances the generalization and adaptability of our model. Structurally, we increase the model's representation of complex patterns by stacking multiple spatial-temporal blocks. Extensive experiments on 4 real-world datasets demonstrate that our method outperforms other baseline methods. In the future, we will delve deeper into learning stable representations through spatial-temporal pre-training, which will further inform the modeling of long-range temporal dependencies and dynamic spatial correlations.

7 Declarations

Competing interests. The authors have no competing interests to declare that are relevant to the content of this paper.

Authors' contributions. Xunlian Luo, Chunjiang Zhu, and Detian Zhang conceived of the presented idea. Xunlian Luo and Chunjiang Zhu wrote the main manuscript text. Xunlian Luo carried out the experiment. Xunlian Luo, Chunjiang Zhu, Detian Zhang and Qing Li reviewed the manuscript. Detian Zhang supervised the project.

Funding. Chunjiang Zhu is supported by UNCG Start-up Funds and Faculty First Award. Detian Zhang is partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization, the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Availability of data and materials. The traffic speed datasets are provided by the open source work DCRNN(https://github.com/liyaguang/DCRNN).

References

[1] Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B.: Deep learning on traffic prediction: Methods, analysis, and future directions. IEEE Trans. Intell. Transp. Syst. **23**(6), 4927–4943 (2022)

[2] Guo, S., Lin, Y., Wan, H., Li, X., Cong, G.: Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Transactions on Knowledge and Data Engineering, 5415–5428 (2021)

[3] Tedjopurnomo, D.A., Bao, Z., Zheng, B., Choudhury, F.M., Qin, A.K.: A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. IEEE Transactions on Knowledge and Data Engineering, 1544–1561 (2020)

[4] Li, H., Zhao, Y., Mao, Z., Qin, Y., Xiao, Z., Feng, J., Gu, Y., Ju, W., Luo, X., Zhang, M.: A survey on graph neural networks in intelligent transportation systems. CoRR **abs/2401.00713** (2024)

[5] Jiang, R., Yin, D., Wang, Z., Wang, Y., Deng, J., Liu, H., Cai, Z., Deng, J., Song, X., Shibasaki, R.: Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction. In: CIKM, pp. 4515–4525 (2021)

[6] Ye, J., Zhao, J., Ye, K., Xu, C.: How to build a graph-based deep learning architecture in traffic domain: A survey. IEEE Trans. Intell. Transp. Syst. **23**(5), 3904–3924 (2022)

[7] Luo, X., Zhu, C., Zhang, D., Li, Q.: Stg4traffic: A survey and benchmark of spatial-temporal graph neural networks for traffic prediction. CoRR **abs/2307.00495** (2023)

[8] Li, F., Feng, J., Yan, H., Jin, G., Yang, F., Sun, F., Jin, D., Li, Y.: Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. ACM Transactions on Knowledge Discovery from Data, 1–21 (2023)

[9] Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, pp. 3634–3640 (2018)

[10] Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. In: IJCAI, pp. 1907–1913 (2019)

[11] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots:

Multivariate time series forecasting with graph neural networks. In: SIGKDD, pp. 753–763 (2020)

[12] Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: 6th International Conference on Learning Representations, ICLR (2018)

[13] Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. In: NeurIPS, pp. 17804–17815 (2020)

[14] Weng, W., Fan, J., Wu, H., Hu, Y., Tian, H., Zhu, F., Wu, J.: A decomposition dynamic graph convolutional recurrent network for traffic forecasting. Pattern Recognit. **142**, 109670 (2023)

[15] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)

[16] Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., Li, P.: DSTAGNN: dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In: International Conference on Machine Learning, ICML 2022. Proceedings of Machine Learning Research, vol. 162, pp. 11906–11917 (2022)

[17] Cirstea, R., Yang, B., Guo, C., Kieu, T., Pan, S.: Towards spatio- temporal aware traffic time series forecasting. In: 38th IEEE International Conference on Data Engineering, ICDE, pp. 2900–2913 (2022)

[18] Shao, Z., Zhang, Z., Wang, F., Wei, W., Xu, Y.: Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting. In: CIKM, pp. 4454–4458 (2022)

[19] Wang, Z., Nie, Y., Sun, P., Nguyen, N.H., Mulvey, J.M., Poor, H.V.: ST-MLP: A cascaded spatio-temporal linear framework with channel-independence strategy for traffic forecasting. CoRR **abs/2308.07496** (2023)

[20] Kumar, S.V., Vanajakshi, L.: Short-term traffic flow prediction using seasonal arima model with limited input data. European Transport Research Review, 1–9 (2015)

[21] Wu, C., Ho, J., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)

[22] Lartey, B., Homaifar, A., Girma, A., Karimoddini, A., Opoku, D.: Xgboost: a tree-based approach for traffic volume prediction. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1280–1286 (2021)

[23] Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional

and recurrent networks for sequence modeling. CoRR (2018)

[24] Fu, R., Zhang, Z., Li, L.: Using lstm and gru neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 324–328 (2016)

[25] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS (2017)

[26] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)

[27] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Networks Learn. Syst., 4–24 (2021)

[28] Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. Expert Systems with Applications, 117921 (2022)

[29] Fang, Z., Long, Q., Song, G., Xie, K.: Spatial-temporal graph ODE networks for traffic flow forecasting. In: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 364–373 (2021)

[30] Choi, J., Choi, H., Hwang, J., Park, N.: Graph neural controlled differential equations for traffic forecasting. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI, pp. 6367–6374 (2022)

[31] Jiang, J., Han, C., Zhao, W.X., Wang, J.: Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: AAAI, pp. 4365–4373 (2023)

[32] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.*: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)

[33] Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: CVPR, pp. 1954–1963 (2021)

[34] Huang, S., Wang, D., Wu, X., Tang, A.: Dsanet: Dual self-attention network for multivariate time series forecasting. In: Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E.A., Carmel, D., He, Q., Yu, J.X. (eds.) CIKM, pp. 2129–2132 (2019)

[35] Liu, H., Dong, Z., Jiang, R., Deng, J., Deng, J., Chen, Q., Song, X.: Staeformer: Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting. CoRR **abs/2308.10425** (2023)

[36] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L.: Transformers

in time series: A survey. In: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, pp. 6778–6786

[37] Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., Jiang, M.: A survey of knowledge-enhanced text generation. ACM Comput. Surv. **54**(11s), 227–122738 (2022)

[38] Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: NeurIPS, pp. 577–585 (2015)

[39] Lin, H., Bai, R., Jia, W., Yang, X., You, Y.: Preserving dynamic attention for long-term spatial-temporal prediction. In: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 36–46 (2020)

[40] Ye, X., Fang, S., Sun, F., Zhang, C., Xiang, S.: Meta graph transformer: A novel framework for spatial-temporal traffic prediction. Neurocomputing **491**, 544–563 (2022)

[41] Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499 (2016)

[42] Dauphin, Y.N., Fan, A., Auli, M., Grangier, D.: Language modeling with gated convolutional networks. In: International Conference on Machine Learning, pp. 933–941 (2017)

[43] Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., Yan, X.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. NeurIPS (2019)

[44] Shao, W., Jin, Z., Wang, S., Kang, Y., Xiao, X., Menouar, H., Zhang, Z., Zhang, J., Salim, F.: Long-term spatio-temporal forecasting via dynamic multiple-graph attention. In: IJCAI (2022)

[45] Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction. In: AAAI, pp. 1234–1241 (2020)

[46] Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z.: Freeway performance measurement system: mining loop detector data. Transportation research record **1748**(1), 96–102 (2001)

[47] Jin, D., Shi, J., Wang, R., Li, Y., Huang, Y., Yang, Y.: Trafformer: Unify time and space in traffic prediction. In: Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI, pp. 8114–8122 (2023)

[48] Shin, Y., Yoon, Y.: PGCN: progressive graph convolutional networks for spatial-temporal traffic forecasting. IEEE Trans. Intell. Transp. Syst. **25**(7), 7633–7644 (2024)