Journal of Artificial Intelligence Research, vol. 2, no.1, pp. 1-8, 2025

STGL: Self-Supervised Spatio-Temporal Graph Learning for Traffic Forecasting

Zhe Zhan[a], Xingtian Mao[b], Hui Liu[a,*] and Shuo Yu[a]

[a]Dalian University of Technology, Dalian, China
[b]BYD Company Limited, Shenzhen, China

ARTICLE INFO

Keywords:
spatio-temporal graph
self-supervised learning
traffic forecasting
contrastive learning

ABSTRACT

As urbanization intensification, traffic forecasting emerges as a critical challenge due to the complex spatio-temporal dependencies and data scarcity in traffic networks. Although spatio-temporal graph neural networks (STGNNS) have demonstrated certain efficacy, these methods cannot effectively model the complex characteristics of traffic data. Meanwhile, their performance is also constrained by the limited data volume and the scarcity of labels. To solve the aforementioned issues, we propose STGL, a self-supervised spatio-temporal graph learning framework for traffic forecasting. STGL utilizes a dual-module architecture to effectively model complex spatio-temporal dependencies in traffic data. Specifically, it integrates a dynamic graph convolution module to capture evolving spatial dependencies, and it utilizes a temporal convolution module leveraging dilated causal convolutions and gated mechanisms to model long-range temporal dependencies. To further enhance representation learning, STGL incorporates a contrastive learning with sample generation and negative filtering. By combining these components, STGL provides a robust solution for traffic forecasting under data-short conditions. Extensive experiments are conducted on PEMS04 and PEMS08, which shows the superiority of STCL.

1. Introduction

As urbanization accelerates, traffic forecasting has become a critical challenge within intelligent transportation systems (ITS) [1]. Traffic forecasting aims to predict the traffic information in the future, such as flow, speed, and congestion levels based on historical and real-time data. Accurate traffic forecasting is crucial for logistics distribution, urban planning, and improving the efficiency of transportation systems [2]. Moreover, the advent of autonomous driving technology introduces additional complexity, as it demands robust traffic prediction capabilities in intricate urban road environments [3]. This underscores the growing importance of advancing traffic forecasting methodologies to support the evolution of intelligent mobility solutions. However, traffic data is inherently complex, characterized by dynamic spatio-temporal dependencies and significant variability [4]. These complexities make it difficult for traditional methods to capture the intricate patterns required for precise predictions.

Spatio-Temporal Graph Neural Networks (STGNNs) are a powerful tool to sovle the above challenges. STGNNs combine graph convolutional networks (GCNs) with recurrent convolutional networks (RNN) or convolutional neural networks (CNN) to model the spatial and temporal dimensions of traffic data.[5]. Despite their successes, STGNNs still face two major challenges. First, current STGNNs have insufficient capability in modeling traffic data [6]. They struggle to effectively model the dynamic nature of traffic data, often relying on static graph structures that fail to adapt to changing traffic patterns. Second, the performance of these models is constrained by the limited availability of labeled data, which is costly to acquire and often scarce in real-world scenarios [7]. Furthermore, the data scarcity issue has become increasingly prominent in recent years, as data privacy has garnered greater attention.

To address these limitations, this paper proposed STGL, a novel self-supervised spatio-temporal graph learning framework. STGL leverages a dual-module architecture to enhance the modeling of dynamic spatio-temporal dependencies. Specifically, it integrates a dynamic graph convolution module to capture evolving spatial dependencies. In addition, it utilizes a temporal convolution module leveraging dilated causal convolutions. We also integrate gated mechanisms into the temporal convolution module to model long-range temporal dependencies. To further enhance representation learning, STGL incorporates a contrastive learning framework to reduce dependency on labeled data. STGL employs diverse data augmentation techniques, such as edge masking and input shielding, to generate positive and negative samples. Additionally, it filters hard negative samples based on temporal correlations, thereby enhancing the discriminative power for learned embeddings. Extensive experiments on PEMS04 [8] and PEMS08 [9] demonstrate that STGL outperforms the state-of-the-art baselines in terms of prediction accuracy and robustness. The contributions of this paper are listed as follows.

- We propose STGL, a dual-module architecture that effectively captures dynamic spatio-temporal dependencies in traffic data. By integrating dynamic graph convolution layer and a temporal convolution layer

DOI: https://doi.org/10.70891/JAIR.2025.040001
ISSN of JAIR: 3078-5529

*Corresponding author
zhanzhe0514@gmail.com (Z. Zhan); xunyinlk@gmail.com (X. Mao); liuhui1126@dlut.edu.cn (H. Liu); yushuo@dlut.edu.cn (S. Yu)

with dilated causal convolutions, STGL is able to model both evolving spatial relationships and long-range temporal dependencies.

- We introduce a contrastive learning pipeline that enhances model performance. STGL employs a rich set of data augmentation methods, including edge masking and input shielding, which reduces reliance on labeled data and improves feature discriminability.
- The effectiveness of STGL is demonstrated through comprehensive experiments, showing significant improvements over existing methods on benchmark datasets such as PEMS04 and PEMS08.

2. Related Works

2.1. Traffic Forecasting

As a key component of ITS, traffic forecasting has been studied for a long time. The integrated auto-regressive moving average (ARIMA) and Kalman filters are the statistical approaches in the field of traffic forecasting. [10]. Subsequently, many variant of ARIMA and nonparametric methods are applied on this task, including KARIMA [11], ARIMAX [12], k-NN [13], Bayesian network [14], and so on. Following the parametric techniques, many researchers shift their attention to machine learning approaches. In the study by Wu [15], a novel model utilizing Convolutional Neural Network (CNN) is developed, focusing on effectively extracting spatial-related features. It also utilizes the Gated Recurrent Unit (GRU) to process the temporal features. In addtion to CNN, Long Short-Term Memory (LSTM) shows an excellent effect on traffic forecasting. Lu [16] integrates a multicast convolutional block with a stacked LSTM block, focusing on exploring the spatial dependencies inherent in traffic data. Furthermore, considering the impact of rain on traffic flow, Jia [17] processes the rainy climate influences by Deep Belief Network and LSTM. Recently, some Graph Nerual Networks (GNNs) models have been adopted in traffic forecasting tasks. For example, Li [18] proposes DCRNN which combined diffusion process and gated recurrent unit . Yu [19] proposes STGCN which employed a generalization of Chebnet to capture the spatial correlations of traffic data. Kong [20] and Fang [21] combine GNN with Transformer to capture global and local multi-level knowledge. However, despite their advancements, these methods often struggle to effectively model the complex spatial dependencies inherent in traffic networks.

2.2. Spatio-Temporal Graph Neural Networks

STGNNs has been proved as a promising approach for traffic forecasting. STGNNs integrate graph convolutional networks with RNN or CNN, aiming to explore the spatio-temporal dependencies in traffic data. Early works [18, 19] utilized fully convolutional structures to model spatio-temporal patterns. Subsequent studies have explored various innovations, including the use of attention mechanisms, multi-graph structures, and hybrid models to enhance performance [22]. For instance, the MSSTGNN [23] model addresses the limitations of static graph structures by constructing adaptive dynamic graphs from multiple perspectives. Additionally, other approaches like A3T-GCN [24] and GMAN [25] have incorporated attention network to better grasp the complex spatial dependencies and time-varying features inherent in traffic data. Based on above, STDN [26] combines dynamic graphs, spatio-temporal embeddings, and trend-seasonality decomposition to enhance traffic flow forecasting with improved accuracy and reduced computational cost. Despite these advancements, current STGNNs still face challenges in dynamically modeling traffic data. Many methods rely on static graph structures, which fail to adapt to changing traffic patterns. Moreover, the rarity of labeled data remains a significant issue, limiting the performance of these models.

3. Method

3.1. Overview

In this section, we present the framework of STGL model. As illustrated in Figure 1, the STGL model consists of four components: data augmentation, STG Encoder, STG Decoder and loss . The data is augmented by four methods for data diversity. The STG Encoding Module is purposed for capturing the intricate spatial and temporal dependencies within traffic data via the integration of spatial and temporal convolution operations. The encoded features are then passed to the STG Decoder, which leverages these features to generate predictions for future traffic conditions. Simultaneously, a contrastive learning framework is integrated to enhance the ability of STCL to learn discriminative features by distinguishing between positive and negative sample pairs. This dual-branch structure allows the model to efficiently leverage both labeled and unlabeled data, thus enhancing its effectiveness in traffic forecasting tasks.

3.2. Spatio-Temporal Graph Construction

We model the traffic network as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where nodes $\mathcal{V}$ represent intersections or exits and edges $\mathcal{E}$ represent the roads connecting them. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ is constructed based on the geographical distances and flow similarities between nodes. For each time step t, the graph $\mathcal{G}$ is associated with a dynamic feature matrix $\mathbf{X}_t \in \mathbb{R}^{N\times D}$, which captures the traffic conditions at that time. This spatio-temporal graph structure serves as the foundation for our model to capture the complex relationships in traffic data.

To effectively capture the spatial dependencies and dynamically learn the hidden spatial relationships, we create an adaptive adjacency matrix for graph convolution layer. Contrary to conventional approaches that depend on predefined adjacency matrices, our approach learns an adaptive adjacency matrix $\tilde{\mathbf{A}}_{\text{adp}}$ through random gradient descent. This matrix is derived from two learnable node embedding dictionaries $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{N\times C}$, as shown in Eq 1. The adaptive adjacency matrix not only reflects the direct and indirect relationships between nodes but also adapts dynamically to

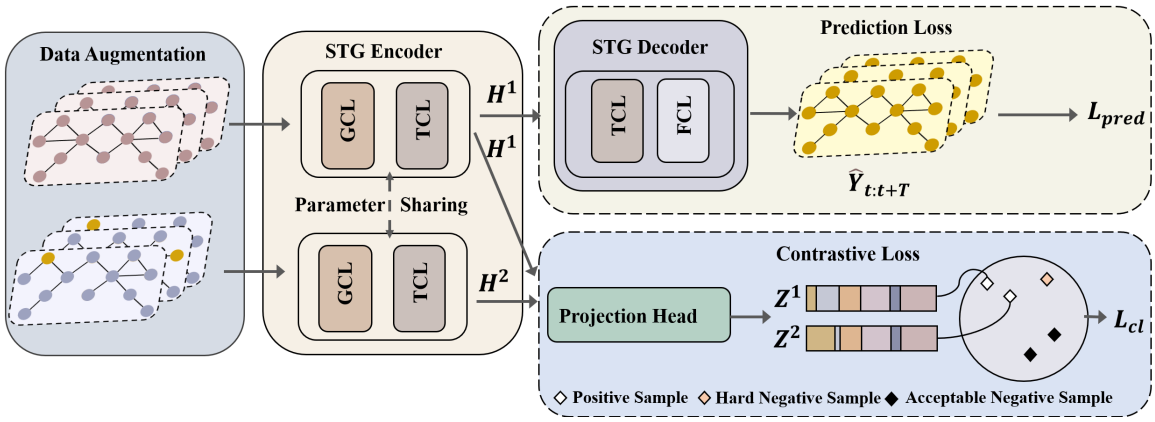

Figure 1: The framework of STGL.

the input data features, capturing spatial dependencies in traffic data.

$$\tilde{\mathbf{A}}_{\text{adp}} = \text{Softmax}(\text{ReLU}(\mathbf{E}_1\mathbf{E}_2^T)) \tag{1}$$

3.3. Data Augmentation

Data augmentation is crucial for improving the generalization ability and robustness of the model, especially in the context of spatio-temporal graph data. We propose several data augmentation methods tailored for traffic prediction tasks, which introduce reasonable disturbances to the graph structure and node features while preserving the essential spatio-temporal relationships. These techniques boost diversity of training data and enable the model to acquire more robust and distinctive features.

3.3.1. Edge Perturbation

Edge perturbation is designed to modify the graph structure by randomly adding or removing edges from the adjacency matrix. However, in the case of weighted adjacency matrices commonly used in spatio-temporal graphs, directly adding edges may not be straightforward. We employ a revised method where we mask elements of the adjacency matrix, thereby altering the graph structure. The augmented adjacency matrix $\mathbf{A}'$ is generated as follows:

$$A'_{ij} = \begin{cases} A_{ij}, & M^E_{ij} \geq r_{em} \\ 0, & M^E_{ij} < r_{em} \end{cases} \tag{2}$$

where $\mathbf{M^E} \sim \mathcal{U}(0,1)$ is a random matrix and r_{em} is an adjustable parameter that controls the proportion of edges to be perturbed. This technique can be applied to both predefined and adaptive adjacency matrices, effectively adding structural diversity to the graph.

3.3.2. Input Masking

Input masking is utilized to boost the model's ability to handle missing values We achieve this by randomly masking certain entries of the feature matrix $\mathbf{X}$. Specifically, each entry of the feature matrix $\mathbf{P}^{(t-S):t}$ is generated as:

$$P^{(t-S):t}_{ij} = \begin{cases} X^{(t-S):t}_{ij}, & M^I_{ij} \geq r_{im} \\ -1, & M^I_{ij} < r_{im} \end{cases} \tag{3}$$

where $\mathbf{M^I} \sim \mathcal{U}(0,1)$ is a random matrix and r_{im} is an adjustable parameter that determines the masking ratio. In this way, the model learns to generate predictions even when some input features are missing, thereby improving its reliability in practical applications.

3.3.3. Temporal Translation

Inspired by the continuous nature of spatio-temporal graph data, which is often sampled at discrete time intervals, we introduce temporal translation as a data augmentation technique. This method leverages the intermediate states between consecutive time steps by performing linear interpolation. The implementation is as follows:

$$P^{(t-S):t} = \alpha X^{(t-S):t} + (1-\alpha)X^{(t-S+1):(t+1)} \tag{4}$$

where α is generated from a uniform distribution $\mathcal{U}(r_{\text{ts}}, 1)$ and r_{ts} is an adjustable parameter. This approach allows the model to delve deeper into temporal dynamics and boosts its capability to track how traffic patterns change over time.

3.3.4. Input Smoothing

In order to decrease the effect of noise in spatio-temporal graph data, we implement input smoothing within the frequency domain. Specifically, the historical data and future values are concatenated, extending time series to $L = S+T$, resulting in $\mathbf{X}^{(t-S):(t+T)} \in \mathbb{R}^{L\times N}$. Then, a discrete cosine transform (DCT) is applied to convert the sequence of node from time domain to frequency domain. The high-frequency components $L - E_{is}$ are scaled down while keeping the low-frequency components E_{is} unchanged. The scaling process involves the following steps: First, generate a random matrix $\mathbf{M} \in \mathbb{R}^{(L-E_{is})\times N}$ from a uniform distribution $\mathcal{U}(r_{is}, 1)$, where r_{is} is an adjustable parameter. Next, smooth the generated matrix using the normalized adjacency matrix $\tilde{\mathbf{A}}$ by

computing $\mathbf{M} = \mathbf{M}\tilde{\mathbf{A}}^2$. This step leverages the idea that adjacent sensors should have similar scaling ranges. If the adjacency matrix is unavailable, this step can be omitted. Last, element-wise multiply the random numbers with high-frequency components $L - E_{is}$. Finally, an inverse DCT is applied to transform the data back to the time domain.

$$\mathbf{M} = \mathbf{M}\mathbf{A}^2 \tag{5}$$

After scaling, the data is transformed back to the time domain using an inverse DCT. This method helps the model focus on the low-frequency components that carry more significant information while reducing the influence of high-frequency noise.

These data augmentation methods collectively contribute to the ability of STCL to learn from diverse and realistic scenarios, making it more robust and accurate in traffic prediction tasks.

3.4. STG Encoder & Decoder

In this section, we will introduce the STG Encoder and Decoder. The encoder, with graph convolution layer (GCL) and temporal convolution layer (TCL), captures spatial and temporal dependencies. The GCL uses diffusion convolution to handle multi-hop neighbor info. In addition, the TCL uses dilated causal convolutions and gating to model long-term patterns. Next, we will focus on the STG Decoder, which uses a TCL and a fully connected layer to map the encoder output to the prediction space.

3.4.1. STG Encoder

The encoder is composed of graph convolution layers and temporal convolution layers.

The graph convolution layers model the spatial dependencies by aggregating and transforming node features according to the graph structure. Among these, the diffusion convolution layer [27] is particularly important as it models the diffusion process on the graph, capturing the multi-hop neighbor information and the underlying spatial dependencies more effectively. As mentioned in Sec3.2, we utilize the ReLU activation function to remove weak connections and employ the SoftMax function to standardize the adaptive adjacency matrix. Consequently, the normalized adaptive adjacency matrix may be viewed as the transition matrix of an diffusion process. By integrating the predefined spatial dependencies with the self-learned latent graph dependencies, the graph convolution layer is formulated as follows:

$$Z = \sum_{k=0}^{K} \left(P_f^k X W_{k1} + P_b^k X W_{k2} + \tilde{A}_{adp}^k X W_{k3} \right) \tag{6}$$

where $\mathbf{P}$ is the transition matrix, K is the number of diffusion steps, $\mathbf{W}^{(l)}$ is the weight matrix for the l-th layer, and σ is a non-linear activation function. This operation allows the encoder to capture the spatial dependencies over multi-hop nodes, providing a more comprehensive understanding of the spatial relationships in the traffic network.

In addition to graph convolution, the encoder also utilizes temporal convolution layers to model the temporal dynamics and capture long-range temporal dependencies. Specifically, we use dilated causal convolution to enlarge the receptive field, and gated temporal convolutional networks to enhance the modeling of temporal sequences [28]. For dilated causal convolution layer, suppose the input sequence is $X \in \mathbb{R}^{T \times D}$, where T is the number of time steps and D is the feature dimension at each time step. Let the filter size be K and the dilation factor be d. The output sequence $Y \in \mathbb{R}^{T \times C}$ is computed as:

$$Y_t = \sum_{s=0}^{K-1} f_s \cdot X_{t-d \cdot s} \tag{7}$$

where, t is the current time step, d is the dilation factor, f_s denotes the weights of the filter, and $X_{t-d \cdot s}$ represents the features at time step $t - d \cdot s$. For gated Temporal convolutional layer, After we get the output Y from the dilated causal convolution, we process Y through the gated temporal convolutional network.

$$H_t = g(W_1 \cdot Y_t + b_1) \odot \sigma(W_2 \cdot Y_t + b_2) + c \tag{8}$$

where W_1, W_2 are the weight matrices of the convolutional filters, b_1, b_2 are the bias vectors, c is the bias term, g is the activation function (e.g., tanh), σ is the sigmoid function.

3.4.2. STG Decoder

The STG Decoder is composed of two main components: the TCL and a fully connected layer (FCL). As the TCL has been detailed in the encoder section, we will only briefly mention it here. The fully connected layer serves as the subsequent component, which takes the output from the TCL and transforms it into the desired output format. This layer is crucial for mapping the extracted features to the target prediction space, ensuring that the final output aligns with the required dimensions and characteristics for the specific prediction task.

3.5. Loss

The loss function in the STGL combines the contrastive loss and the supervised loss.

The contrastive loss aims to enhance the ability of STCL to distinguish between positive and negative samples in the embedding space. It is defined as:

$$L_{cl} = \frac{1}{M} \sum_{i=1}^{M} -\ln \frac{\exp(\text{sim}(z_i^1, z_i^2)/\tau)}{\sum_{\substack{j=1 \\ j \neq i}}^{M} \exp(\text{sim}(z_i^1, z_j^2)/\tau)} \tag{9}$$

where τ denotes the temperature parameter. For node/graph i, a total of $M - 1$ negatives are incorporated. Finally, the acquired representations can be utilized for downstream tasks.

The supervised loss, commonly a regression - based function like MSE or MAE, assesses the difference between

model predictions and ground - truth labels. For instance, the MSE loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{10}$$

where y_i is the ground truth value, $\hat{y}_i$ is the predicted value, and N is the number of samples.

The total loss function is a weighted sum of these two losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{supervised}} + \lambda \cdot \mathcal{L}_{\text{contrastive}} \tag{11}$$

where λ is a weighting parameter which balances the contributions of the supervised and contrastive losses. The comprehensive - based loss function allows the model to learn from both the labeled data and the underlying data distribution, thus enhancing its generalization and predictive performance.

4. Experiments

4.1. Experimental Settings

4.1.1. Datasets

The experiments use the PEMS04 and PEMS08 datasets, which contain traffic data collected from sensors on Californian highways. As shown in Table 1, PEMS04 covers 307 nodes and 340 edges with 16,992 time steps, while PEMS08 includes 170 nodes and 295 edges with 17,856 time steps. The data is divided into training, validation, and testing sets in a 7:1:2 ratio.

4.1.2. Baselines

The baseline models selected for comparison are Graph WaveNet [9], DCRNN [18], STGCN [19], AGCRN [29], and STFGNN [30]. Graph WaveNet integrates adaptive adjacency matrices with graph convolutions and utilizes 1D convolutions for time series processing. DCRNNformulates graph convolutions via a diffusion process and combines GCN with recurrent models in an encoder-decoder architecture for multi-step prediction. STGCN, a spatio-temporal graph convolutional network, deploys GCN layers and temporal convolutional layers to capture spatial and temporal correlations. AGCRN enhances graph convolutions with two adaptive modules and integrates them into an RNN. STFGNN fuses spatial and temporal graphs, combining the fusion module with a new gated convolutional module in a unified layer. These models represent state-of-the-art approaches in traffic forecasting and provide a robust benchmark for evaluating the performance of the proposed STGL model.

4.1.3. Evaluation Metrics

STGL is assessed by three key metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). The three metrics quantify the deviation between predicted and actual traffic flow values, with lower values indicating better performance.

Table 1
Dataset introduction.

Dataset	Nodes	Edges	Timesteps
PEMS04	307	340	16,992
PEMS08	170	295	17,856

Table 2
Evaluation metrics data for the PEMS04 dataset in STGL and baseline models.

Model	MAE	MAPE (%)	RMSE
Graph WaveNet	20.64	14.66	32.72
DCRNN	21.21	14.85	33.56
STGCN	21.90	15.41	35.96
AGCRN	21.32	14.92	35.00
STFGNN	20.32	14.32	32.24
STGL	**20.06**	**14.12**	**31.92**

4.1.4. Experiment Parameters

All experiments are carried out using a personal computer with a nvidia-4090 GPU. All models are implemented using PyTorch 3.10, except for STFGNN which used MXNet. The batch size is set to 64, the learning rate to 0.001, and the number of training epochs to 100.

4.2. Experimental Results

4.2.1. Traffic Forecasting

As shown in Table 2 and Table 3, STGL demonstrates superior performance compared to baseline models on both PEMS04 and PEMS08 datasets. On PEMS04, STGL achieved the lowest MAE, MAPE, and RMSE values. Similar results are observed on PEMS08, where STGL outperformed all baseline models. Specifically, in terms of the three evaluation metrics of MAE, MAPE, and RMSE, the STGL model records values that are respectively 0.26, 0.2, and 0.32 lower than the optimal baseline values on PEMS04 dataset, and 0.81, 0.11, and 0.43 lower on PEMS08 dataset. The lower values of MAE, MAPE, and RMSE correspond to a higher level of model accuracy. Consequently, these findings highlight the effectiveness of the STGL model in traffic forecasting.

In order to more comprehensively analyze the performance of the STGL model, we visualizes and compares the predicted and ground truth values of node 0 in the test set. As shown in Figure 2, the STGL model can capture the dynamic traffic flow characteristics quite well. Although the STGL model performs well in capturing extended dependencies and periodic fluctuations in time series data, it still encounters challenges in responding to rapid changes in traffic data. When traffic flow fluctuates within a short time, the model may fail to capture these changes in a timely and accurate manner, leading to increased prediction errors. Therefore, improving the model's responsiveness to rapid changes in traffic data will be a key focus of future research.

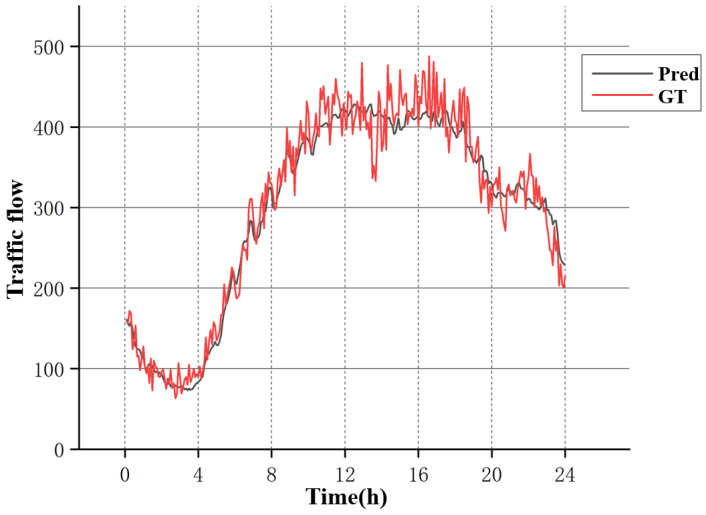

Figure 2: Visualization of prediction (Pred) and ground truth (GT).

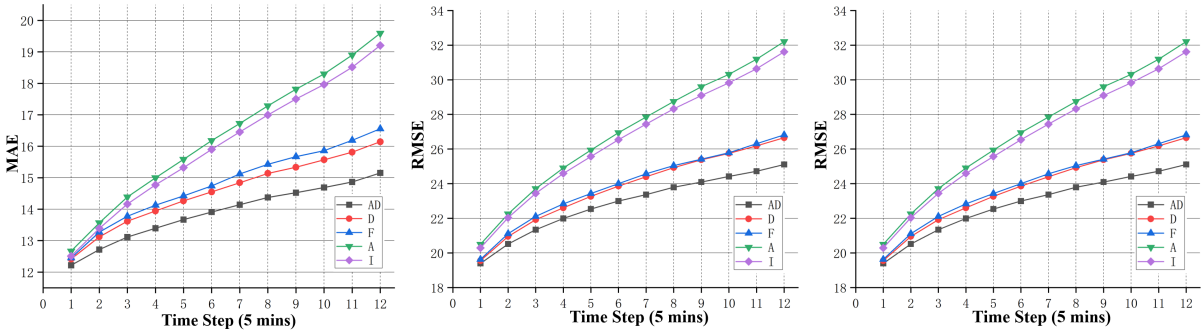

Figure 3: Ablation experiments on PEMS08 for MAE, MAPE and RMSE metircs.

Table 3
Evaluation metrics data for the PEMS08 dataset in STGL and baseline models.

Model	MAE	MAPE (%)	RMSE
Graph WaveNet	15.85	10.56	25.66
DCRNN	17.04	10.69	26.46
STGCN	18.96	11.29	28.52
AGCRN	18.50	11.57	29.23
STFGNN	17.09	10.82	26.67
STGL	**15.04**	**10.45**	**25.23**

4.2.2. Ablation Study

To verify the influence of the adaptive adjacency matrix and diffusion convolution layer on the STGL model's performance, ablation studies are conducted on PEMS08 dataset.

As shown in Figure 3, we present the MAE, MAPE, and RMSE metrics for the STGL model's predictions on the first 12 time steps (5–60 minutes) of the PEMS08 dataset under these configurations. The configurations are represented as follows: I represents training with only the predefined adjacency matrix; F and D denote forward and bidirectional diffusion; A indicates the adaptive adjacency matrix alone; and AD combines bidirectional diffusion with the adaptive adjacency matrix. Analysis of the experimental results shows that both the adaptive adjacency matrix and diffusion convolution positively contribute to the model's performance. Models trained with the adaptive adjacency matrix alone show minimal improvement over those using only the predefined matrix. However, configurations with forward or bidirectional diffusion outperform the baseline model. Furthermore, combining diffusion configurations with the adaptive adjacency matrix leads to the best performance. This demonstrates that the adaptive adjacency matrix enhances the ability of STCL to capture spatial relationships, while diffusion convolution effectively models temporal dependencies, together significantly improving the model's accuracy in traffic forecasting.

4.2.3. Parameter Sensitivity Analysis

This section aims to evaluate the impact of different parameters in data augmentation techniques and the negative sample filtering threshold. For these experiments, the MAE metric from the PEMS08 dataset is employed as the evaluation criterion. To evaluate the effectiveness of various data augmentation methods under different parameters, each augmentation method is individually tested with fine-tuned

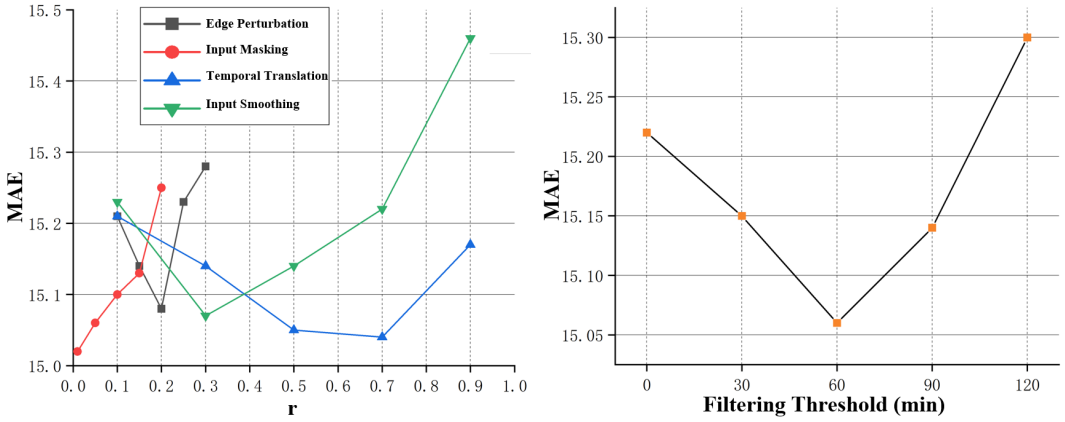

Figure 4: Experiment results across different augmentation techniques and filtering thresholds.

hyperparameters, and the results are systematically analyzed as shown in Figure 4.

The figure on the left illustrates the outcomes of these experiments. Compared with Table 3, we can get the follow conclusions: Notably, despite the semantic differences among the augmentation methods, the performance disparities between the best achievements of each method are marginal. This indicates that the framework exhibits low sensitivity to the semantic variations of the proposed augmentation methods. This could be attributed to the framework's stronger emphasis on temporal and spatial relationships rather than specific image semantics. Next, we observe that the input shielding method is highly sensitive to perturbation magnitude, whereas other augmentation methods demonstrate relative stability with no significant trends. This sensitivity may stem from the substantial perturbation caused by shielding a certain proportion of entries to zero, whereas perturbations from other methods are deemed reasonable by the model. Consequently, the selection of the input shielding method requires careful consideration based on specific circumstances and task requirements.

The figure on the right presents the effects of the filtering threshold r_f, based on the temporal correlations of traffic data. The experiments adopts a default setting of 1% input shielding. The results indicate that setting the filtering threshold r_f to 60 minutes yields the best model performance, validating the effectiveness of the proposed solution. However, when r_f is set to 120 minutes, the results deteriorate compared to when r_f is 0. This may be due to the excessive filtering of negative samples at 120 minutes, weakening the contrastive learning task. In such cases, the model may fail to acquire discriminative knowledge beneficial for the prediction task. Therefore, filtering out the most hard negative negative samples can focus the contrastive loss on true negatives. Howvever, over-filtering can undermine the contrastive task and lead to performance declines.

5. Conclusion

In this paper, we propose STGL, a novel self-supervised spatio-temporal graph learning framework for traffic forecasting. STGL can successfully capture complex spatio - temporal relationships in traffic data via its two - module structure. This structure integrates a dynamic graph convolution module with a temporal convolution module that uses dilated causal convolutions and gated mechanisms. Furthermore, STGL integrates a contrastive learning framework that uses data augmentation techniques and negative sample filtering to reduce reliance on labeled data and enhance feature discriminability. Various Experiments are conducted on PEMS04 and PEMS08 datasets. It demonstrates that STGL outperforms state-of-the-art baseline models. The results highlight the effectiveness of STGL in handling the challenges of traffic forecasting, particularly under data-scarce conditions.

Despite these achievements, we acknowledge that there are still opportunities for further improvement. Future research directions will concentrate on refining the model's ability to respond to rapid fluctuations in traffic data [31]. Additionally, we plan to apply STGL into large-scale traffic scenarios and improve its computational efficiency to meet the demands of real-time traffic prediction [32]. We also plan to explore the integration of multi-modal data into the STGL framework to enrich the traffic information captured by the model. Overall, we believe that STGL lays a solid foundation for advancing the field of traffic forecasting and will keep refining this framework to meet the evolving needs of intelligent transportation systems.

Acknowledgement

The authors declare that there is no funding and no conflict of interest.

References

[1] Q. Yuan, J. Wang, Y. Han, Z. Liu, W. Liu, DAGCAN: decoupled adaptive graph convolution attention network for traffic forecasting, IEEE Trans. Intell. Transp. Syst. 26 (2025) 3513–3526.

[2] W. Xiong, R. Fonod, A. Alahi, N. Geroliminis, Multi-source urban traffic flow forecasting with drone and loop detector data, CoRR abs/2501.03492 (2025).

[3] C. Kim, H. Yoon, S. Seo, S. Kim, STFP: simultaneous traffic scene forecasting and planning for autonomous driving, in: International Conference on Intelligent Robots and Systems,, IEEE, 2021, pp. 6016–6022.

[4] A. Ali, I. Ullah, S. Ahmad, Z. Wu, J. Li, X. Bai, An attention-driven spatio-temporal deep hybrid neural networks for traffic flow prediction in transportation systems, IEEE Transactions on Intelligent Transportation Systems (2025).

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

[6] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, B. Du, Adaptive spatio-temporal graph neural network for traffic forecasting, Knowledge-Based Systems 242 (2022) 108199.

[7] L. Liu, Y. Yu, Y. Wu, Z. Hui, J. Lin, J. Hu, Method for multi-task learning fusion network traffic classification to address small sample labels, Scientific Reports 14 (2024) 2518.

[8] J. Ma, Z. Guo, S. Shi, H. He, N. Duffield, Traffic prediction with diverse historical memory: A multi-component rnn approach, in: 2019 IEEE 35th International Conference on Data Engineering, IEEE, 2019, pp. 1234–1245.

[9] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2019, pp. 1914–1920.

[10] A. Emami, M. Sarvi, S. A. Bagloee, Short-term traffic flow prediction based on faded memory kalman filter fusing data from connected vehicles and bluetooth sensors, Simulation Modelling Practice and Theory 102 (2020) 102025.

[11] M. Van Der Voort, M. Dougherty, S. Watson, Combining kohonen maps with arima time series models to forecast traffic flow, Transportation Research Part C: Emerging Technologies 4 (1996) 307–318.

[12] B. M. Williams, Multivariate vehicular traffic flow prediction: evaluation of arimax modeling, Transportation Research Record 1776 (2001) 194–200.

[13] G. A. Davis, N. L. Nihan, Nonparametric regression and short-term freeway traffic forecasting, Journal of transportation engineering 117 (1991) 178–188.

[14] S. Sun, C. Zhang, G. Yu, A bayesian network approach to traffic flow forecasting, IEEE Transactions on intelligent transportation systems 7 (2006) 124–132.

[15] Y. Wu, H. Tan, L. Qin, B. Ran, Z. Jiang, A hybrid deep learning based traffic flow prediction method and its understanding, Transportation Research Part C: Emerging Technologies 90 (2018) 166–180.

[16] H. Lu, D. Huang, Y. Song, D. Jiang, T. Zhou, J. Qin, St-trafficnet: A spatial-temporal deep learning network for traffic forecasting, Electronics 9 (2020) 1474.

[17] W. Zhang, R. Li, P. Shang, H. Liu, Impact analysis of rainfall on traffic flow characteristics in beijing, International Journal of Intelligent Transportation Systems Research 17 (2019) 150–160.

[18] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: 6th International Conference on Learning Representations, OpenReview.net, 2018.

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

[20] J. Kong, X. Fan, M. Zuo, M. Deveci, X. Jin, K. Zhong, Adct-net: Adaptive traffic forecasting neural network via dual-graphic cross-fused transformer, Information Fusion 103 (2024) 102122.

[21] Y. Fang, Y. Liang, B. Hui, Z. Shao, L. Deng, X. Liu, X. Jiang, K. Zheng, Efficient large-scale traffic forecasting with transformers: A spatial data management perspective, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1, KDD '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 307–317.

[22] J. Gao, C. Guo, Y. Liu, P. Li, J. Zhang, M. Liu, Dynamic-static feature fusion with multi-scale attention for continuous blood glucose prediction, in: Proceedings of the 2025 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, Hyderabad, India, 2025, pp. 1–5.

[23] Z. Chen, C. Wang, J. Zhang, H. Li, F. Wang, Msstgnn: Multi-scaled spatio-temporal graph neural networks for traffic prediction, Knowledge-Based Systems 295 (2024) 110880.

[24] J. Zhu, Y. Song, L. Zhao, H. Li, A3t-gcn: Attention temporal graph convolutional network for traffic forecasting, Applied Sciences 10 (2020) 2485.

[25] Y. Zheng, J. Li, Z. An, Gman: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, AAAI Press, 2019, pp. 1201–1208.

[26] L. Cao, B. Wang, G. Jiang, Y. Yu, J. Dong, Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, 2025, pp. 11463–11471.

[27] Y. Li, R. Yu, C. Shahabi, Y. Liu, U. Demiryurek, R. Zhang, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations, 2018.

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

[29] L. Bai, L. Yao, C. Li, H. Yu, W. Kang, X. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, in: Advances in Neural Information Processing Systems, volume 33, 2020, pp. 17804–17815.

[30] M. Li, Z. Zhu, Spatial-temporal fusion graph neural network for traffic flow forecasting, in: AAAI Conference on Artificial Intelligence, 2021.

[31] J. Xue, S. Sun, M. Liu, Y. Wang, X. Meng, J. Wang, J. Zhang, K. Xu, Burst-sensitive traffic forecast via multi-property personalized fusion in federated learning, IEEE Transactions on Mobile Computing (2025) 1–17.

[32] Y. Yan, S. Cui, J. Liu, Y. Zhao, B. Zhou, Y.-H. Kuo, Multimodal fusion for large-scale traffic prediction with heterogeneous retentive networks, Information Fusion 114 (2025) 102695.