Progressive Dynamic Graph Sparsification for Spatiotemporal Traffic Forecasting

Yiren Zhou
College of Computer Science
Sichuan University
Chengdu, China
zhouyiren1@stu.scu.edu.cn

Shiyong Lan
College of Computer Science
Sichuan University
Chengdu, China
lanshiyong@scu.edu.cn

Yao Ren
College of Computer Science
Sichuan University
Chengdu, China
renyao1@stu.scu.edu.cn

Zicheng Sun
College of Computer Science
Sichuan University
Chengdu, China
2024223045034@stu.scu.edu.cn

***Abstract*—Accurate traffic forecasting is the cornerstone of Intelligent Transportation Systems (ITS), yet it remains a formidable challenge due to the complex non-linear spatiotemporal dependencies. Existing methods often struggle to balance physical topological constraints with dynamic semantic correlations. Furthermore, temporal modeling frequently employs fully connected structures or standard attention mechanisms, which lead to noise propagation from irrelevant historical time steps and lack multi-scale hierarchy. To address these issues, we propose a novel framework named PDGCN (Progressive Dynamic Graph Sparsification Network). Spatially, we introduce a Dual-View Fusion mechanism that combines static physical diffusion with dynamic semantic interaction to capture complementary spatial dependencies. Temporally, we design a Progressive Temporal Graph Sparsification strategy. By utilizing a hierarchical Top-K masking mechanism, the model adaptively filters temporal noise, achieving a curriculum learning process from High-Salience Core Features to Global Comprehensive Semantics. Experiments on real-world datasets (PeMS03, PeMS04, PeMS07, and PeMS08) demonstrate that PDGCN achieves state-of-the-art performance, and ablation studies validate the effectiveness of the proposed sparsification strategy. Source codes are available at https://github.com/porridgea/PDGCNporridgea/PDGCN temporally.**

***Index Terms*—Traffic Forecasting, Graph Neural Networks, Dynamic Sparsification, Spatiotemporal Modeling, Curriculum Learning**

I. Introduction

Accurate traffic flow prediction lies at the core of Intelligent Transportation Systems (ITS), providing a vital foundation for traffic management, route planning, congestion warning, and smart city construction [1]. Accurate traffic forecasting hinges on the effective capture of intricate spatiotemporal correlations. However, traffic flow exhibits significant dynamics and complex coupling characteristics in spatiotemporal dimensions: spatially, there are local influences propagating along the physical road network as well as potential cross-regional long-range correlations; temporally, it involves both periodic changes and short-term fluctuations [2]–[4]. Effectively capturing these non-linear, multi-scale spatiotemporal correlations remains a central challenge in this field.

In recent years, Spatio-Temporal Graph Neural Networks (STGNNs) have reshaped this domain. From early methods based on predefined static graphs (e.g., STGCN [3], DCRNN [4],ASTGCN [5]) to dynamic graph methods introducing adaptive adjacency matrices (e.g., Graph WaveNet [6], AGCRN [7],MTGNN [8], ST-AWGNN [9]), spatial modeling capabilities have continuously improved. Recently, Transformer-based architectures (e.g., PDFormer [10], TimeMixer [11], STAEformer [12], STP-FPTN [13]) have established new performance benchmarks by leveraging their ability to capture long-range dependencies. However, existing methods still face two fundamental bottlenecks:

First, regarding spatial modeling, existing research fails to effectively coordinate local physicality with global semantics. Traditional static graph methods [3], [4] can capture local physical diffusion using predefined road network topologies, but they are powerless against implicit global dependencies within the road network. Conversely, while existing adaptive graph learning methods are flexible [6], [7], most adopt a parameterized generation approach—i.e., generating a static adaptive graph by learning fixed node embeddings [8]. This method faces two problems: first, it is *pseudo-dynamic*, as the graph structure remains fixed during inference and cannot respond to real-time traffic perturbations [14]; second, it faces the risk of learning spurious correlations due to over-parameterization [15]. Lacking physical constraints, it easily learns false global connections on noisy data [15], thereby ignoring the fundamental objective laws of traffic propagation along the physical road network [14]. This binary opposition between local physical constraints and global semantic exploration remains largely unresolved [16]–[19].

Second, regarding temporal modeling, current State-of-the-Art (SOTA) methods (especially Transformer-based classes) generally adopt fully connected temporal attention mechanisms [10], implicitly assuming that the current state is correlated with all moments in the historical window. However, traffic flow data is inherently sparse and highly salient in terms of temporal correlation—prediction results are often determined by a few key moments (e.g., the same period of the previous day, periodic peak hours, or recent trends) rather than the entire observation window. Such a fully connected structure not only incurs computational redundancy [20], but also introduces substantial irrelevant historical information as noise. As the network depth increases, the propagation of this noise leads to feature homogenization—commonly referred to as the *oversmoothing* phenomenon [21], [22]—thereby severely impairing the model's discriminative capability.

Addressing the above challenges, based on the ideas of Multi-View Complementary Learning [23] and Curriculum Learning [24], this paper proposes **PDGCN** (Progressive Dynamic Graph Convolutional Network). In spatial modeling, we design a dual-view spatial module to achieve dynamic-static complementarity and a balance between local and global perspectives: (1) The local static branch performs graph diffusion based on distance-constrained road network topology to capture deterministic geographical proximity constraints [3]; (2) The global dynamic branch innovatively introduces a data-driven dynamic graph generation mechanism, dynamically inferring long-range semantic dependencies between nodes based on real-time traffic flow data [7]. In temporal modeling, unlike existing methods relying solely on single vanilla self-attention, we introduce a progressive temporal graph sparsification strategy as a core innovative component. Working in parallel with the standard temporal modeling branch and inspired by curriculum learning, this strategy guides the model from focusing on shallow high-saliency core moments to gradually transitioning to deep global semantic integration through a layer-wise evolving Top-K mask.

The main contributions of this paper are summarized as follows:

- We propose a **Dual-View Spatial Modeling** mechanism that effectively fuses static physical diffusion with dynamic semantic graph learning. This complementary design achieves comprehensive spatial dependency modeling without sacrificing physical interpretability or modeling flexibility.
- We introduce a **Progressive Temporal Graph Sparsification** strategy, a novel curriculum-style structural denoising method inspired by human cognitive learning processes. By adaptively filtering irrelevant temporal connections via layer-wise Top-K masking and evolving sparsity levels, we introduce curriculum learning into graph structure denoising for the first time. This hierarchical sparsity evolution effectively mitigates noise propagation and oversmoothing in deep networks.
- Extensive experiments on four real-world benchmark datasets demonstrate that PDGCN consistently outperforms state-of-the-art baselines.

II. Methodology

A. Problem Definition

We define the traffic road network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ is the set of N sensor nodes, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the predefined adjacency matrix encoding the physical connection relationships. Let $\mathbf{X}_t \in \mathbb{R}^{N \times C}$ denote the network-wide traffic state at time step t, where C is the number of features. We organize the historical observations of a batch into a 4D tensor $\mathcal{X}_{in} \in \mathbb{R}^{B \times C \times N \times T_{in}}$, where B denotes the batch size. To assist temporal modeling, we also introduce auxiliary temporal embeddings $\mathbf{T}_{emb} \in \mathbb{R}^{T_{in} \times D}$ (e.g., time-of-day, day-of-week). The goal of spatiotemporal traffic forecasting is to learn a mapping function f. Given the historical observation tensor, the physical graph structure and the auxiliary temporal embeddings, we predict the traffic states for the future T_{out} time steps:

$$\hat{\mathcal{X}}_{out} = f(\mathcal{G}, \mathbf{T}_{emb}; \mathcal{X}_{in}) \tag{1}$$

B. Framework Overview

As shown in Fig. 1, we first project the raw input into a d-dimensional hidden representation:

$$\mathcal{H}^{(0)} = \text{Proj}(\mathcal{X}_{in}) \in \mathbb{R}^{B \times d \times N \times T} \tag{2}$$

PDGCN comprises L identical spatiotemporal encoding layers. At layer l, the input tensor $\mathcal{H}^{(l-1)}$ is processed in parallel by three functional branches, whose outputs are then aggregated via a gated fusion module. The three branches are defined as follows:

- **Dual-View Spatial Modeling Branch:** Includes a local static spatial branch and a global dynamic spatial branch, jointly modeling local physical constraints and long-range semantic dependencies.
- **Curriculum-Guided Complementary Temporal Branch:** (i) a Progressive Temporal Branch that employs layer-wise sparsification to denoise historical signals, and (ii) a Global Temporal Branch that maintains long-range continuity to capture background trends.
- **Auxiliary Feature Projection Branch:** Enhances inter-channel interactions and preserves the fidelity of the original signal.

C. Dual-View Spatial Modeling

To reconcile the rigid constraints of physical topology with the flexibility of semantic associations, we design a dual-view spatial module.

1) Local Static View: The evolution of traffic flow is strictly constrained by the physical road network topology (e.g., upstream flow inevitably affects downstream). This branch aims to capture this deterministic traffic diffusion process. By enforcing locality, we leverage physical inductive bias to prevent the model from overfitting to spurious long-range correlations in the early stages of training.

a) Distance-Threshold-Based Graph Reconstruction: Original adjacency matrices often contain noise or miss multi-hop influences. We propose a physical distance-based hard-thresholding strategy. Using the Dijkstra algorithm to calculate the shortest path distance $dist(v_i, v_j)$ between nodes, we select a threshold ϵ based on the statistical distribution of road network distances to reconstruct a sparse adjacency matrix $\mathbf{A}^{phy}$:

$$\mathbf{A}_{ij}^{phy} = \begin{cases} 1, & \text{if } dist(v_i, v_j) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

This operation structurally severs nodes that are physically distant but might have noisy connections in the raw data, providing a denoised physical topology prior.

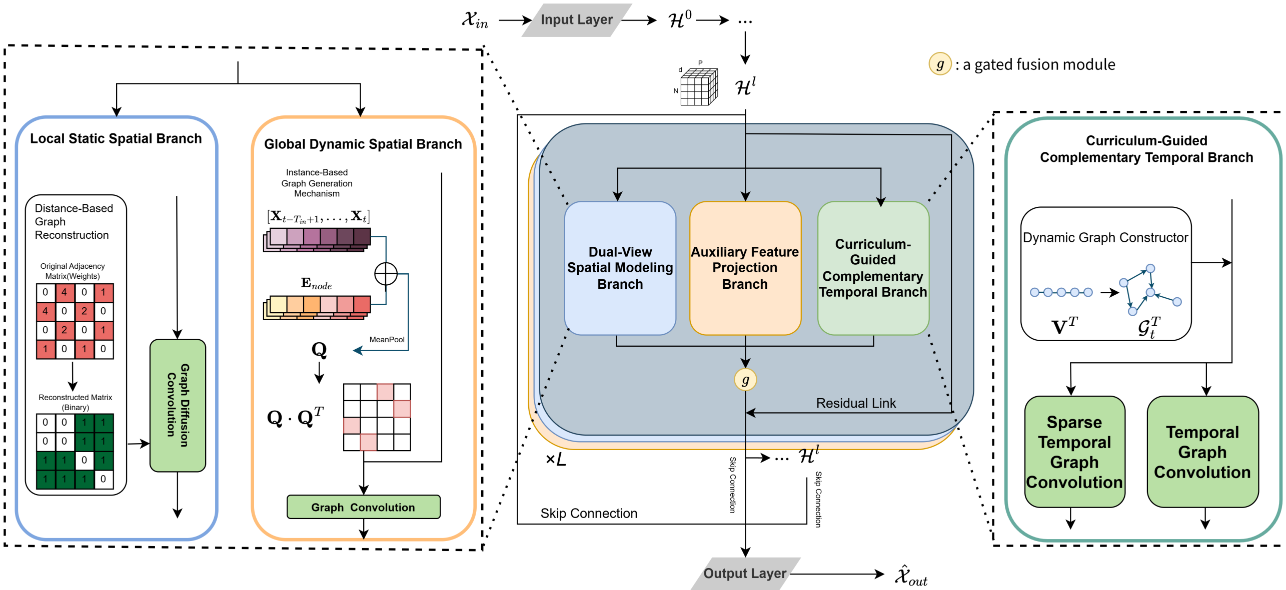

Fig. 1. The overall architecture of PDGCN. The model consists of L identical stacked layers. The input $\mathbf{X}_{t-T_{in}+1:t}$ is transformed into the initial state $\mathcal{H}^0$, which is then processed by three modules in each layer: (1) The **Dual-View Spatial Module** treats road segments as nodes to model spatial relations via static and dynamic graph convolutions; (2) The **Curriculum-Guided Complementary Temporal Module** treats time slots as nodes to capture temporal patterns via dense and sparse graph convolutions; (3) A fusion module aggregates features for the next layer. Finally, skip connections from all layers are utilized to predict traffic flow for the future T_{out} time steps.

b) Graph Diffusion Convolution: We model traffic flow as a random walk diffusion process on the reconstructed physical graph. Let $\tilde{\mathbf{A}} = \mathbf{A}^{phy} + \mathbf{I}_N$ be the adjacency matrix with self-loops, and $\mathbf{D}$ be the corresponding degree matrix. We define the random-walk transition (diffusion) matrix as

$$\mathbf{P} = \mathbf{D}^{-1}\tilde{\mathbf{A}} \tag{4}$$

The local static features $\mathcal{H}_{static}^{(l)}$ at layer l are computed by

$$\mathcal{H}_{static}^{(l)} = \mathrm{ReLU}\Big(\mathbf{P}\,\mathcal{H}^{(l-1)}\,\mathbf{W}_s\Big) \tag{5}$$

where $\mathbf{W}_s$ is a learnable weight matrix. Since $\mathbf{A}^{phy}$ is constructed by a distance threshold ϵ, the diffusion operator $\mathbf{P}$ only propagates messages along physically plausible edges. Therefore, the effective receptive field of node v_i after l stacked diffusion layers is confined within its l-hop neighborhood on $\mathcal{G}^{phy}$:

$$ERF^{(l)}(v_i) \subseteq \mathcal{N}_{\leq l}^{phy}(v_i) \tag{6}$$

Equivalently, any node reached within l hops corresponds to a physical path composed of edges whose lengths are bounded by ϵ, which enforces a strong locality inductive bias and preserves physically consistent diffusion dynamics.

2) Global Dynamic View: To break the physical receptive field limitations of the local static branch to capture long-range semantic dependencies, and to resolve the pseudo-dynamic defect of existing adaptive graph methods (which rely on fixed parameter matrices and cannot respond to real-time traffic perturbations), we propose the Global Dynamic View. Unlike traditional methods that learn static global relationships, this module adopts an instance-based graph generation mechanism, dynamically generating the adjacency matrix directly based on the input features of the current time window.

a) Instance-Based Graph Generation Mechanism: Traditional adaptive graph methods typically generate the graph structure $\mathbf{A} \sim \mathbf{E}_1\mathbf{E}_2^T$ by training two fixed node embedding matrices $\mathbf{E}_1, \mathbf{E}_2$. This essentially captures the average relationship across all time steps in the training set, ignoring dynamic changes within specific time windows. To address this, we shift the source of graph generation from fixed parameters to real-time data. First, we introduce a learnable node embedding $\mathbf{E}_{node} \in \mathbb{R}^{1\times d\times N\times T}$ and add it to the hidden state to supplement implicit static attributes:

$$\mathcal{Z}^{(l)} = \mathcal{H}^{(l-1)} + \mathbf{E}_{node} \tag{7}$$

To extract spatial descriptors representing the specific characteristics of the nodes in the current layer, we perform average pooling across the temporal dimension:

$$\mathbf{Q}^{(l)} = \mathrm{MeanPool}_T(\mathcal{Z}^{(l)}) \in \mathbb{R}^{B\times d\times N} \tag{8}$$

This design is crucial: $\mathbf{Q}^{(l)}$ varies with the layer input $\mathcal{H}^{(l-1)}$. When unexpected conditions occur in the road network, the feature vectors in $\mathbf{Q}^{(l)}$ change instantly, triggering an adaptive adjustment of the graph structure. Then, the instance-based adjacency matrix $\mathbf{A}_{dyn}^{(l)}$ is calculated via a self-attention mechanism [25]:

$$\mathbf{A}_{dyn}^{(l)} = \mathrm{Softmax}\left(\mathrm{ReLU}\left(\frac{\mathbf{Q}^{(l)}\cdot(\mathbf{Q}^{(l)})^\top}{\sqrt{d}}\right)\right) \tag{9}$$

where d is a normalization coefficient. This dense matrix allows any two nodes to directly exchange information based

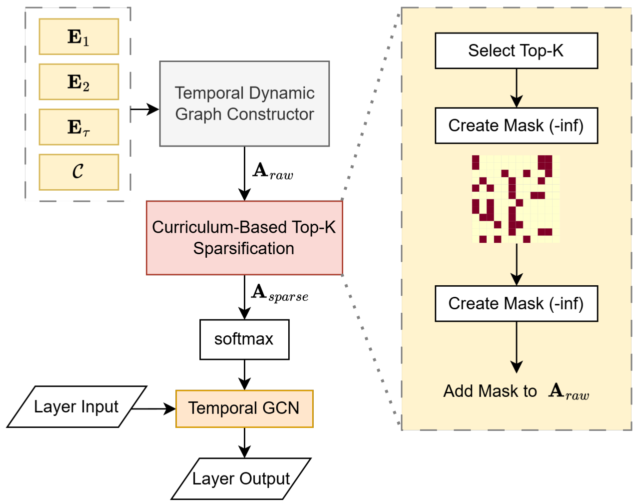

Fig. 2. The detailed structure of the Sparse Temporal Graph Convolution module. It utilizes Tucker Tensor Decomposition to compute the initial temporal affinity, followed by a Curriculum-based Top-K Masking mechanism to filter out irrelevant historical time steps. The resulting sparse adjacency matrix is then used to perform temporal graph convolution.

on semantic similarity within the current time window, structurally achieving a global receptive field.

b) Semantic Feature Propagation: Based on the generated $\mathbf{A}_{dyn}^{(l)}$, we perform graph convolution operations to aggregate global context information:

$$\mathcal{H}_{dynamic}^{(l)} = \text{Dropout}(\mathbf{A}_{dyn}^{(l)} \mathcal{H}^{(l-1)}) \tag{10}$$

Finally, the global semantic features extracted by this branch are fused with the local static features to form a complete spatial representation.

D. Progressive Temporal Branch

Traditional temporal attention mechanisms typically construct fully connected temporal dependency graphs, assuming that current traffic states are equally important to all historical moments. However, traffic flow data often exhibits burstiness and specific periodicity, meaning the state at a certain moment is usually strongly correlated with only a few key historical moments. A fully connected structure not only introduces computational redundancy but also aggregates irrelevant time steps as noise, leading to the *over-smoothing* phenomenon in deep network features. Existing sparsification methods (e.g., fixed windows or random DropEdge) lack adaptability. Inspired by the human cognitive process—from point to surface, gradually transitioning from capturing salient features to understanding complex global patterns—we propose a curriculum-based progressive temporal graph sparsification mechanism as illustrated in Fig. 2. This module adopts a *structure denoising* strategy, implementing a progressive expansion of the receptive field from small to large by dynamically filtering irrelevant time steps.

1) Temporal Dynamic Graph Constructor: To capture the latent dependencies between historical time steps while maintaining parameter efficiency, we employ a graph generator based on Tucker Tensor Decomposition [26]. Unlike standard attention mechanisms that utilize static parameters, our approach incorporates dynamic temporal contexts via a batch-wise generation mechanism.

Specifically, we define two types of learnable embeddings: (1) Sequence Step Embeddings $\mathbf{E}_1^{(l)}, \mathbf{E}_2^{(l)} \in \mathbb{R}^{T\times d_e}$, representing the relative positional information of the T historical steps; and (2) Time-of-Day Embeddings $\mathbf{E}_\tau^{(l)} \in \mathbb{R}^{N_\tau \times d_e}$, capturing the absolute semantic context of the time slices (where N_τ is the total number of time slots per day, and d_e is the embedding dimension). We also introduce a learnable core tensor $\mathcal{C}^{(l)} \in \mathbb{R}^{d_e\times d_e\times d_e}$ to model the interaction patterns [27].

For a batch of input samples with size B at time indices $\mathbf{t} \in \mathbb{R}^B$, the dynamic temporal adjacency tensor $\mathbf{A}_{raw}^{(l)} \in \mathbb{R}^{B\times T\times T}$ is computed as:

$$\mathbf{A}_{raw}^{(l)} = \text{ReLU}\left(\mathcal{C}^{(l)} \times_1 \mathbf{E}_\tau^{(l)}[\phi(\mathbf{t})] \times_2 \mathbf{E}_1^{(l)} \times_3 \mathbf{E}_2^{(l)}\right) \tag{11}$$

where $\phi(\mathbf{t})$ maps the current batch of time indices to their daily slots, yielding a batch-specific embedding tensor of size $B \times d_e$. The operator $\times_n$ denotes the mode-n tensor product. This generates unique temporal graphs for each sample in the batch, allowing the model to adaptively distinguish patterns between different times of the day.

2) Curriculum-Based Top-K Sparsification: We formalize the sparsification process as a curriculum learning problem. Define the curriculum schedule $\mathcal{S} = \{(l, k_l)\}_{l=1}^{L}$, where l is the layer index and k_l is the number of temporal neighbors retained in the l-th layer. We adopt a progressive linear growth strategy to control the sparsity $\rho_l = k_l/T$:

$$k_l = k_{\min} + \left\lceil \frac{(k_{\max} - k_{\min}) \cdot (l-1)}{L-1} \right\rceil \tag{12}$$

For the l-th layer, we generate a binary mask tensor $\mathbf{M}^{(l)} \in \mathbb{R}^{B\times T\times T}$. Specifically, for the b-th sample in the batch and time step i, we identify the set of indices $\Omega_{b,i}^{(l)}$ corresponding to the Top-k_l largest affinity values in the i-th row of the generated adjacency tensor $\mathbf{A}_{raw}^{(l)}$:

$$\Omega_{b,i}^{(l)} = \text{argtopk}(\mathbf{A}_{raw}^{(l)}[b, i, :], k_l) \tag{13}$$

The mask entries are then defined as follows to filter out non-Top-K connections:

$$\mathbf{M}_{b,i,j}^{(l)} = \begin{cases} 0 & \text{if } j \in \Omega_{b,i}^{(l)} \\ -\infty & \text{otherwise} \end{cases} \tag{14}$$

3) Sparse Temporal Graph Convolution: The final sparse temporal adjacency matrix $A_{sparse}^{(l)}$ is obtained by applying the mask and performing Softmax normalization. This ensures that only the most relevant temporal connections participate in feature updates, utilizing the values after ReLU activation as intensity weights:

$$\mathbf{A}_{sparse}^{(l)} = \text{Softmax}(\mathbf{A}_{raw}^{(l)} + \mathbf{M}^{(l)}) \tag{15}$$

Finally, graph convolution is performed based on this sparse dynamic graph to update temporal features:

$$\mathcal{H}_{progressive}^{(l)} = \text{Dropout}(\mathbf{A}_{sparse}^{(l)} \mathcal{H}^{(l-1)}) \tag{16}$$

This mechanism effectively executes explicit structural denoising, improving the model's robustness to sudden traffic noise, while utilizing multi-granular sparse structures to achieve hierarchical feature extraction.

E. Global Temporal Branch

To compensate for the potential loss of long-range continuity information caused by the hard truncation in the progressive branch, we design a parallel global temporal branch aiming to capture smooth background trends via a full view. This branch follows the Temporal Dynamic Graph Constructor described previously (Section II-D). We use a separate set of parameters for this branch (including ${\mathbf{E}'_1}^{(l)}, {\mathbf{E}'_2}^{(l)}, \mathbf{E}'^{(l)}_{\tau}$ and a global core tensor $\mathcal{C}^{(l)}_{global}$)to reconstruct the raw temporal affinity $\mathbf{A}'^{(l)}_{raw}$.

The core difference from the progressive branch is that this branch does not impose any structural sparsity mask but directly applies Softmax normalization to the raw scores:

$$\mathbf{A}^{(l)}_{global} = \text{Softmax}(\mathbf{A}'^{(l)}_{raw}) \tag{17}$$

The resulting $\mathbf{A}^{(l)}_{global}$ is a dense tensor, ensuring non-zero information pathways between any time steps within the historical window. Subsequently, we leverage this dense graph to aggregate contextual features: $\mathcal{H}^{(l)}_{global} = \text{Dropout}(\mathbf{A}^{(l)}_{global}\mathcal{H}^{(l-1)})$. The outputs of the progressive branch and the global branch are finally concatenated along the channel dimension, thereby achieving comprehensive coverage of both salient sparse patterns and global background trends.

F. Other Components and Training Process

The Auxiliary Feature Projection Branch acquires auxiliary features $\mathcal{H}^{(l)}_{feat}$ by applying a linear fully connected layer across the feature dimension. To integrate multi-view information from the local static, global dynamic, progressive temporal, global temporal, and auxiliary branches, we adopt a gating fusion strategy.

First, we concatenate the outputs from all branches along the channel dimension:

$$\mathcal{H}^{(l)}_{cat} = [\mathcal{H}^{(l)}_{static} \parallel \mathcal{H}^{(l)}_{dynamic} \parallel \mathcal{H}^{(l)}_{prog} \parallel \mathcal{H}^{(l)}_{global} \parallel \mathcal{H}^{(l)}_{feat}] \tag{18}$$

Then, the final output $\mathcal{H}^{(l)}$ of the l-th layer is obtained through the following residual structure:

$$\mathcal{H}^{(l)} = \text{BatchNorm}\left(\mathcal{H}^{(l-1)} + \phi_{fuse}\left(\text{ReLU}(\mathcal{H}^{(l)}_{cat})\right)\right) \tag{19}$$

where $[\cdot]$ denotes channel concatenation, and ϕ_{fuse} is a 1×1 convolutional layer used for dimensionality reduction and feature reorganization. Moreover, hidden states in every layer are skip connected to the output layer for the final forecasting:

$$\hat{\mathbf{x}}_{i+1:i+T_{out}} = MLP(\sum_{l=1}^{L} \mathcal{H}^{l}) \tag{20}$$

Traffic flow data often contains outliers caused by sensor failures or sudden accidents. Traditional Mean Squared Error (MSE) is highly sensitive to outliers, easily leading to gradient explosion, while Mean Absolute Error (MAE) is non-differentiable at zero, which is unfavorable for model convergence. To balance robustness and optimization stability, we adopt Huber Loss as the objective function. Huber Loss behaves as a quadratic function (MSE) when errors are small, ensuring gradient smoothness; and as a linear function (MAE) when errors are large, effectively reducing the impact of outliers on backpropagation. The objective function is defined as follows:

$$\mathcal{L} = \frac{1}{|\Omega|} \sum_{(i,t)\in\Omega} L_\delta(\hat{x}_{i,t}, x_{i,t}) \tag{21}$$

$$L_\delta(\hat{x}, x) = \begin{cases} \frac{1}{2}(\hat{x} - x)^2, & |\hat{x} - x| \leq \delta, \\ \delta(|\hat{x} - x| - \frac{1}{2}\delta), & |\hat{x} - x| > \delta \end{cases} \tag{22}$$

where $\hat{x}_{i,t}$ and $x_{i,t}$ are the predicted and ground truth values respectively, Ω is the set of valid observations, and δ is a hyperparameter balancing the squared error. The model is trained end-to-end using the Adam optimizer. To stabilize deep network training, we employ Gradient Clipping to restrict the gradient norm. Additionally, a Multi-Step Learning Rate Decay strategy is implemented to facilitate escaping local optima and achieving refined convergence. An Early Stopping mechanism based on validation loss is also adopted to prevent overfitting.

III. Experiments

In this section, we conduct extensive experiments on four real-world public transportation datasets to answer the following three core Research Questions (RQs):

- **RQ1 (Overall Performance):** How does the prediction accuracy of PDGCN compare with existing SOTA models (including classic and latest GNN-based methods)?
- **RQ2 (Ablation Analysis):** What are the specific contributions of the proposed Dual-View Spatial Modeling and Progressive Temporal Graph Sparsification strategies to the model performance?
- **RQ3 (Visual Interpretability):** How does the temporal attention distribution evolve from shallow to deep layers, and does it validate the effectiveness of the proposed curriculum-based structure?

A. Experimental Settings

a) Datasets: We evaluated the model on four standard benchmark datasets from the Caltrans Performance Measurement System (PeMS) [28]: PEMS03, PEMS04, PEMS07, PEMS08. These datasets contain traffic flow data aggregated at 5-minute intervals. The dataset statistics are shown in Table I.

b) Baselines: We compared PDGCN with two groups of representative baseline methods. The first group is classic spatiotemporal methods including STGCN [3], DCRNN [4], Graph WaveNet (GWNet) [6], ASTGCN [5], MTGNN [8],D2STGNN [14], LSGCN [29], STSGCN [30], STPGNN [31], GDGCN [27] and PDFormer [10]. The second group is the latest spatiotemporal models including STDN (AAAI 2025) [32] and STSGNN (AAAI 2025) [33].

TABLE I
DETAILS OF USED DATASETS.

Dataset	Sensors (Nodes)	Time Steps	Time Range
PeMS03	358	26,208	Sep. 2018 – Nov. 2018
PeMS04	307	16,992	Jan. 2018 – Feb. 2018
PeMS07	883	28,224	May 2017 – Aug. 2017
PeMS08	170	17,856	Jul. 2016 – Aug. 2016

c) Parameter Settings: All experiments were conducted on a machine equipped with an NVIDIA GeForce 4090 GPU. We set the input sequence length $T_{in} = 12$ and the prediction length $T_{out} = 12$. The ratio of training, validation, and testing sets across all four datasets is 6:2:2. N_t is set to 2016. The batch size is set to 32. The optimizer used is Adam, training for 300 epochs with a patience of 30. The hidden state dimension d is selected from $\{16, 32, 64, 128, 256\}$. The number of network layers L is selected from $\{4, 5, 6, 7, 8\}$. The learning rate is selected from $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}\}$, and the weight decay coefficient is selected from $\{1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}\}$. A multi-step learning rate decay strategy is adopted, decaying the learning rate to 0.1 times the original at the 100th and 200th epochs. k_{min} is set to 3, and k_{max} is selected from $\{8, 10, 12\}$.

d) Evaluation Metrics: We employ three widely used metrics to evaluate prediction performance: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

B. Overall Performance Comparison (RQ1)

Table II presents the performance comparison between PDGCN and baseline models on the PEMS03, PEMS04, PEMS07, and PEMS08 datasets. We report the average MAE, RMSE, and MAPE for 12-step (60 minutes) prediction.

From Table II, we can observe the following:

1) **SOTA Superiority:** PDGCN outperforms existing SOTA models across almost all metrics on the four datasets. This indicates that PDGCN is capable of more accurately capturing complex spatiotemporal dependencies.
2) **Long-range Modeling Capability:** Compared to traditional GNN-based methods (such as STGCN and DCRNN), PDGCN achieves significant performance improvements on all datasets. This proves that the dual-view spatial module can effectively balance local physical constraints and global semantic dependencies when dealing with large-scale road networks, avoiding the performance degradation seen in traditional methods on large-scale graphs due to restricted information propagation paths.
3) **Noise Robustness:** PDGCN performs exceptionally well on the MAPE metric across all datasets. Since MAPE is more sensitive to outliers and noise, the advantage in this metric suggests that the progressive sparsification strategy effectively filters temporal noise, enhancing the model's robustness against sudden traffic incidents (e.g., accidents, road construction) and sensor failures.
4) **Consistent Performance:** unlike some methods that perform well on specific datasets but differ in performance on others, PDGCN achieves optimal performance on all four datasets, demonstrating good generalization ability and stability.

C. Ablation Study (RQ2)

To verify the effectiveness of individual modules in PDGCN and the rationality of the curriculum learning strategy, we conducted ablation studies on the PeMS04 and PeMS08 datasets. We designed six variants aimed at answering two core questions: (1) Are all spatiotemporal branches necessary? (2) Is the small-to-large progressive sparsification strategy truly effective?

- **w/o Static:** Remove the local static spatial branch.
- **w/o Dynamic:** Remove the global dynamic spatial branch.
- **w/o Global-T:** Remove the global temporal branch.
- **w/o Progressive:** Remove the progressive temporal branch.
- **w/ Fixed Top-K:** The sparsity k per layer is fixed and does not increase with the number of layers.
- **w/ Reversed Top-K:** Reverse the curriculum order, i.e., the sparsity k changes from large to small.

The experimental results are shown in Table III.

1) Necessity of Multi-View Architecture: The first four rows of Table III demonstrate the impact of removing individual branches on model performance:

- **Spatial Complementarity Verification:** Removing either the Local Static Branch or the Global Dynamic Branch led to an increase in prediction error. Notably, the performance degradation caused by removing the dynamic branch was more significant. This suggests that while the physical topology provides necessary spatial constraints, capturing dynamic, data-driven implicit semantic dependencies is more critical when dealing with complex traffic perturbations. The combination of both achieves an optimal balance between physical constraints and semantic flexibility.
- **Importance of Temporal Sparsification:** Among all variants, removing the Progressive Temporal Branch resulted in the most severe performance degradation. This result empirically confirms our core hypothesis: vanilla self-attention mechanisms introduce irrelevant historical information as noise into the computation. Our proposed sparsification strategy effectively acts as a structured filter, significantly enhancing feature purity by retaining only high-value connections.

2) Validation of Curriculum Learning Mechanism: To investigate the specific strategy of sparsification, we compared the standard PDGCN with two strategic variants: fixed Top-K (constant sparsity across layers) and reversed Top-K (sparsity decreasing from large to small, i.e., global to local).

- **Dynamic vs. Static Sparsity:** The fixed Top-K variant performed worse than the full model. This indicates

TABLE II
EVALUATIONS OF PDGCN AND BASELINES ON PEMS03, PEMS04, PEMS07 AND PEMS08.

Dataset	PEMS03			PEMS04			PEMS07			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STGCN	17.49	30.12	17.15%	22.70	35.55	14.59%	25.38	38.78	11.08%	18.02	27.83	11.40%
DCRNN	18.18	30.31	18.91%	24.70	38.12	17.12%	25.30	38.58	11.66%	17.86	27.83	11.45%
GWNet	19.85	32.94	19.31%	25.45	39.70	17.29%	26.85	42.78	12.12%	19.13	31.05	12.68%
ASTGCN	17.69	29.66	19.40%	22.93	35.22	16.56%	28.05	42.57	13.92%	18.61	28.16	13.08%
MTGNN	17.23	25.89	17.35%	19.98	31.92	14.13%	23.92	35.86	12.43%	15.03	23.89	10.23%
D2STGNN	15.10	26.57	15.23%	18.42	29.97	12.81%	19.68	33.24	8.43%	14.35	24.18	9.33%
LSGCN	17.94	29.85	16.98%	21.53	33.86	13.18%	27.31	41.46	11.98%	17.73	26.76	11.20%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
STPGNN	14.87	25.89	15.54%	18.86	30.13	13.14%	21.77	35.28	9.37%	14.69	23.85	9.55%
GDGCN	14.66	24.30	**13.94%**	18.44	29.79	12.52%	20.15	33.21	8.50%	14.82	23.87	9.35%
PDFormer	14.94	25.39	15.82%	18.32	29.96	12.10%	19.83	32.87	8.53%	13.58	23.51	9.05%
STDN	16.05	27.51	17.71%	18.67	30.92	13.16%	22.94	36.06	10.32%	14.79	24.60	10.26%
STSGNN	14.61	25.24	14.43%	18.69	30.84	12.24%	19.87	32.97	8.29%	14.38	23.81	9.25%
PDGCN	**14.37**	**23.52**	14.51%	**18.22**	**29.75**	**12.06%**	**19.59**	**32.56**	**8.21%**	**13.32**	**23.01**	**8.88%**

TABLE III
ABLATION STUDY ON PEMS04 AND PEMS08.

Model	**PeMS04**			**PeMS08**		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PDGCN (Full)	**18.22**	**29.75**	**12.11%**	**13.32**	**23.01**	**8.88%**
w/o Static	18.92	30.56	12.74%	14.02	23.38	9.17%
w/o Dynamic	19.05	30.68	12.89%	14.18	23.45	9.21%
w/o Global-T	18.65	30.08	12.42%	13.71	23.15	9.09%
w/o Progressive	18.78	30.21	12.53%	14.47	23.76	9.48%
w/ Fixed Top-K	18.68	30.35	12.58%	13.57	23.25	8.89%
w/ Reversed Top-K	18.75	30.19	12.45%	13.77	23.28	9.06%

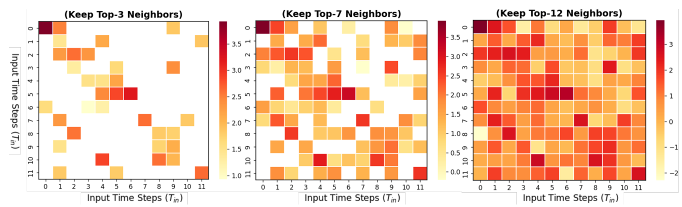

Fig. 3. Visualization of the evolution of temporal attention weights across different layers (Layer 1, 3, and 6) on the PeMS08 dataset. The heatmaps illustrate a clear curriculum learning process: shallow layers focus on sparse, high-saliency local moments (structural denoising), while deep layers gradually expand the receptive field to capture global contextual dependencies.

that a rigid, single-scale receptive field cannot meet the hierarchical feature extraction needs of deep networks. Deep networks intrinsically require shallow layers to focus on local details while deep layers focus on global patterns.

- **Curriculum vs. Anti-Curriculum:** The reversed Top-K variant performed even worse. It forces the shallow layers of the network to process global information containing a large amount of noise prematurely, before robust local features have been established. This contradicts the human cognitive law of learning from simple to complex, from point to surface, causing noise to be amplified and propagated early in the network. This result strongly validates the necessity of our proposed local to global (K increasing) order for effective representation learning.

D. Visual Interpretability (RQ3)

To intuitively understand how the progressive sparsification strategy regulates information flow, we visualized the learned temporal adjacency matrices ($\mathbf{A}_{sparse}^{(l)}$) from Layer 1 (Shallow), Layer 3 (Middle), and Layer 6 (Deep) on the PeMS08 dataset, as shown in Fig. 3.

- **Shallow Layer Behavior:** At Layer 1, the attention weights exhibit high sparsity, concentrating only on adjacent moments and a few significant peak moments. This proves that the model successfully executes structural denoising in the shallow layers.
- **Deep Layer Behavior:** At Layer 6, the attention distribution becomes smoother and broader, covering the entire historical window. This indicates that the deep network is utilizing purified features to integrate global context.

This distinct transition from sparse, high-saliency attention in shallow layers to dense, global attention in deep layers confirms that our strategy successfully enforces a structured learning behavior, aligning with the hierarchical processing nature of human cognition.

IV. CONCLUSION

This paper proposes a Progressive Dynamic Graph Convolutional Network (PDGCN) for traffic flow forecasting. Addressing the issues where existing methods struggle to balance physical and semantic dependencies spatially, and suffer from noise propagation and over-smoothing temporally, we designed a dual-view spatial fusion mechanism and a curriculum learning-based temporal sparsification strategy, respectively. By mimicking the human cognitive process of from point to surface, PDGCN achieves hierarchical feature denoising and extraction. Experiments on four real-world datasets demonstrate that PDGCN significantly outperforms existing SOTA methods. Future work will explore extending this sparsification framework to irregular sampling time series forecasting tasks.

References

[1] B. Medina-Salgado, E. Sánchez-DelaCruz, P. Pozos-Parra, and J. E. Sierra, "Urban traffic flow prediction techniques: A review," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100739, 2022.

[2] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[3] B. Yu, H. Yin, and 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 (IJCAI)*, 2018, pp. 3634–3640.

[4] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations (ICLR)*, 2018.

[5] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[6] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 1907–1913.

[7] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[8] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[9] B. Xu, X. Wang, C. Liu, Z. Liu, and L. Kang, "Dual-channel spatio-temporal wavelet transform graph neural network for traffic forecasting," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[10] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4365–4373.

[11] S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. ZHOU, "Timemixer: Decomposable multiscale mixing for time series forecasting," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=7oLshfEIC2

[12] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, and X. Song, "Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting," in *Proceedings of the 32nd ACM international conference on information and knowledge management*, 2023, pp. 4125–4129.

[13] J. Zhang, J. Tang, J. Jin, and Z. Qu, "Spatio-temporal pre-training enhanced fast pure tansformer network for traffic flow forecasting," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8.

[14] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 2733–2746, 2022.

[15] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4356–4364.

[16] L. Liu, J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin, "Physical-virtual collaboration modeling for intra- and inter-station metro ridership prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3377–3391, 2022.

[17] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 11 906–11 917.

[18] Y. Ren, W. Zhu, S. Lan, X. Zhou, H. Yang, and Z. Hou, "Fe-hgat: Frequency-enhanced hybrid graph attention network for traffic prediction," in *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2025, pp. 5173–5178.

[19] T. Pan, X. Zhou, S. Lan, W. Wang, H. Yang, Z. Li, Z. Hou, and Y. Ren, "Dstfgcn: A dynamic spatial-temporal fusion graph convolution network for traffic flow forecasting," *Neural Networks*, p. 108989, 2026.

[20] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[21] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[22] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations (ICLR)*, 2020.

[23] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[24] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[27] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, and W. Lv, "Generic dynamic graph convolutional network for traffic flow forecasting," *Information Fusion*, vol. 100, p. 101946, 2023.

[28] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation research record*, vol. 1748, no. 1, pp. 96–102, 2001.

[29] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: Long short-term traffic prediction with graph convolutional networks." in *IJCAI*, vol. 7, 2020, pp. 2355–2361.

[30] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 914–921.

[31] W. Kong, Z. Guo, and Y. Liu, "Spatio-temporal pivotal graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 8, 2024, pp. 8627–8635.

[32] L. Cao, B. Wang, G. Jiang, Y. Yu, and J. Dong, "Spatiotemporal-aware trend-seasonality decomposition network for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 463–11 471.

[33] Y. Chen, F. Lin, J. Huo, and H. Yan, "Designing specialized two-dimensional graph spectral filters for spatial-temporal graph modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 500–11 508.