A Periodic Dynamic Graph and Meta-Graph Memory Network for Traffic Flow Prediction

Jianxuan Wei[1] and Chao Shen[1](✉)

[1]School of Information and Electrical Engineering, Hebei University of Engineering, Handan, 056038,HeBei, China
shenchao1010@aliyun.com

Abstract. Forecasting traffic conditions remains a formidable task due to the continuous evolution of spatial correlations and the deep-seated spatiotemporal heterogeneity across road networks. Conventional approaches typically depend on fixed graph structures, making them inadequate for modeling the highly dynamic and cyclical characteristics of city traffic. To bridge this gap, we introduce an innovative architecture termed PDMetaNet. Specifically, we design a Periodic Dynamic Graph Generation Module (PDGM) that creates bidirectional, time-variant adjacency matrices by deeply integrating instantaneous traffic observations with daily and weekly cyclical patterns. Based on this topology, the Periodic Adaptive Graph Convolutional Recurrent Unit (PAGCRU) couples dynamic graph convolutions with gating mechanisms to accomplish joint spatiotemporal feature extraction. Additionally, a Spatio-Temporal Meta-Graph Learner (STMG) is established to construct a memory bank of meta-nodes, which effectively encodes generalized traffic prototypes across varying temporal stages. This querying mechanism permits the network to adaptively retrieve historical experiences to refine the dynamic graph structure. Furthermore, contrastive and consistency constraints are applied to regularize the memory parameters, thereby boosting generalization. Comprehensive evaluations on the PeMS04 and PeMS08 benchmarks reveal that PDMetaNet outperforms ten existing state-of-the-art models, exhibiting remarkable reductions in forecasting errors and exceptional proficiency in modeling complicated traffic dynamics.

Keywords: Intelligent transportation systems, Traffic flow prediction, Meta-graph memory mechanism, Periodic dynamic graph, Graph neural networks.

1 Introduction

The accelerated pace of urban expansion and the corresponding rise in automobile usage have led to severe traffic gridlocks, posing a major threat to the sustainability of metropolitan areas[1][2]. While Intelligent Transportation Systems (ITS) offer viable mitigation strategies, their success in optimizing routes and managing traffic lights depends heavily on the precise estimation of upcoming traffic conditions[3][4]. Because modern road networks exhibit extreme topological complexity and nonlinear variations, deep learning techniques have become the primary solution for traffic prediction tasks. In particular, frameworks that integrate Graph Neural Networks (GNNs) for spatial dependency extraction and Recurrent Neural Networks (RNNs) for temporal sequence modeling have established the current state-of-the-art[5].

In spite of these breakthroughs, accurately profiling the actual spatiotemporal behaviors of vehicular flow is still hindered by several methodological bottlenecks. First, conventional models relying on predefined, static adjacency matrices are structurally inflexible[5][6]. This rigidity means they frequently fail to adapt when road scenarios experience sudden disturbances, such as transient congestion resulting from unexpected traffic collisions. Second, to circumvent the limitations of fixed topologies, adaptive graph learning frameworks have been introduced to dynamically generate spatial dependencies based on real-time traffic inputs[7][8][9]. Nevertheless, these dynamic structures are predominantly reactive to short-term fluctuations. They inherently struggle in complex urban environments because they fail to incorporate the strong periodic regularities of traffic data. Without a proactive mechanism to anticipate cyclic patterns, these models cannot effectively differentiate the distinct spatial interactions occurring during a weekday morning peak from those on a weekend night[10][11]. Finally, while a handful of recent investigations have incorporated external memory banks to tackle spatiotemporal heterogeneity, these components usually function as isolated pattern extractors. They merely refine hidden states in the feature space without establishing a closed-loop feedback system with the graph generation module. Consequently, valuable historical insights and prior knowledge cannot explicitly guide the optimization of the underlying graph topology.

To overcome the rigidities of predefined static graphs and fully exploit the inherent periodic patterns in traffic data, we propose PDMetaNet (Periodic Dynamic Graph with Meta-graph Memory Network). This architecture introduces a suite of synergistic mechanisms tailored to precisely capture complex, time-varying spatiotemporal dependencies. The primary contributions of this paper are summarized as follows:

Pioneering a Periodic Dynamic Graph Generation Module (PDGM). Unlike existing approaches that rely solely on instantaneous states, PDGM utilizes a dynamic-periodic coupling embedding mechanism to seamlessly integrate real-time traffic signals with intra-day and intra-week periodic features. This allows the generated dynamic adjacency matrix to capture both abrupt state fluctuations and long-term periodic evolutionary trends.

Formulating a Spatio-Temporal Meta-Graph Memory Mechanism. Functioning as an empirical knowledge base, this component maintains a learnable bank of meta-nodes to encapsulate and compress prototypical cross-period traffic patterns. Through an attention-based query strategy and a hyper-network architecture, the model adaptively retrieves relevant historical patterns to generate context-specific graph topologies, thereby explicitly addressing spatiotemporal heterogeneity.

Establishing a Closed-Loop Feedback Architecture. We construct a synergistic feedback loop between the periodic dynamic graph and the meta-graph memory. While the dynamic graph supplies the immediate state context for memory querying, the retrieved meta-graphs recursively refine the graph convolutional topology, elegantly balancing real-time adaptability with structural stability derived from historical priors.

Extensive Empirical Validation. Comprehensive experiments conducted on the PeMS04 and PeMS08 datasets demonstrate the superiority of PDMetaNet. Our model consistently outperforms state-of-the-art baselines, achieving an average error reduction of 11.98% across MAE, RMSE, and MAPE metrics, robustly substantiating its practical efficacy for intelligent transportation management.

2 Related Work

Early traffic forecasting relied on statistical models or simple machine learning. With the rise of deep learning, combining GNNs to capture spatial topology with RNNs or Temporal Convolutional Networks (TCNs) for temporal dependencies has yielded state-of-the-art performance[11][12].

Graph structures are vital for capturing road network dependencies. Static graph-based models, such as STGCN[5] and ASTGCN[6], use predefined matrices based on physical or Euclidean distance. To address their rigidity, dynamic graph networks like AGCRN[7] and EvolveGCN[8] adaptively generate topologies driven by real-time node embeddings. However, these methods often focus on capturing instantaneous dynamics without fully leveraging the periodic characteristics of traffic flow.

Given the profound spatiotemporal heterogeneity in traffic networks, some studies have introduced external memory banks or prototype learning to store typical flow patterns[13]. Nevertheless, these mechanisms primarily refine hidden states in the feature space. PDMetaNet differentiates itself by explicitly coupling these historical patterns with the dynamic graph generation process to optimize the structural topology.

3 Preliminaries

In this study, the topological structure of a road network is mathematically represented by an undirected graph G=(V,E) . Here, V encompasses N distinct traffic sensors (regarded as nodes), while E signifies the connecting road segments between them (edges). We use $X_T \in \mathrm{R}^{N\times C}$ to symbolize the observed traffic feature matrix at a specific time interval t , where C indicates the number of measured attributes (e.g., flow, speed). The ultimate goal of the traffic forecasting task is to optimize a predictive mapping function f . This function ingests a sequence of historical observations $x=[X_{t-P+1}, X_{t-P+2}...X_t] \in \mathrm{R}^{N\times P\times C}$ and outputs the future traffic conditions over the next Q horizons, denoted as $y=[X_{t+1}, X_{t+2}...X_{t+Q}] \in \mathrm{R}^{N\times Q\times C}$.

4 Methodology

The architecture of the PDMetaNet model is illustrated in Fig. 1. The model is primarily composed of three core components:

1. A Periodic Adaptive Graph Convolutional Recurrent Unit (PAGCRU) based Encoder-Decoder structure, which serves as the backbone for sequence modeling;
2. A Periodic Dynamic Graph Generation Module (PDGM), responsible for constructing dynamic adjacency matrices fused with periodic priors;
3. A Spatio-Temporal Meta-Graph Learner, designed to retrieve and generate optimized graph topologies from historical patterns.

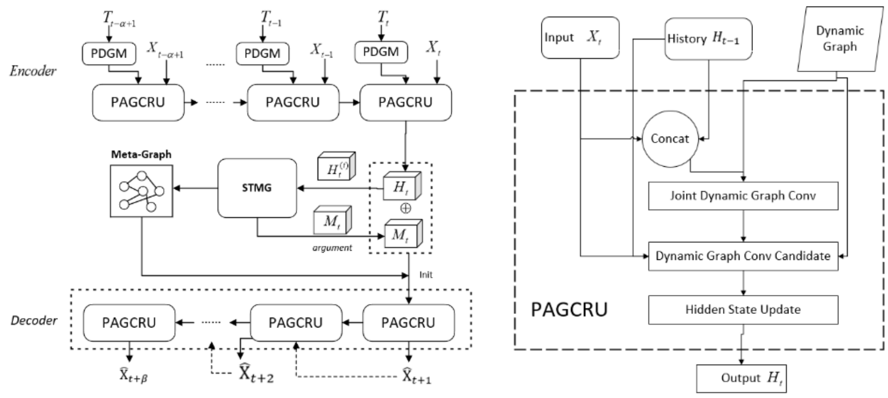

Fig. 1. The structural diagram of the PDMetaNet model. It illustrates the Encoder-Decoder architecture based on PAGCRU, the generation of dynamic graphs via PDGM, and the topological optimization driven by the Spatio-Temporal Meta-Graph Learner.

Traffic interactions across road networks exhibit strong temporal dynamics and periodicity. Consequently, relying on a single static graph derived from physical topology or Euclidean distance is often insufficient. Such static structures fail to adapt to sudden changes in road conditions or align with real-time traffic states. Moreover, they are ill-equipped to capture evolving spatial dependencies or model the distinct characteristics of heterogeneous traffic patterns. To address these issues, we propose the Periodic Dynamic Graph Generation Module (PDGM), which constructs two distinct periodic dynamic graphs.

To account for the recurring behaviors within transportation networks, our framework employs a cyclical embedding strategy aimed at mapping underlying time-based patterns. As demonstrated in Fig. 2, vehicular movements in metropolitan areas display pronounced diurnal and weekly rhythms. To effectively model these dynamics, we establish a pair of distinct learnable memory matrices: $T^D \in \mathbb{R}^{N_d \times p}$ for capturing intra-day fluctuations, and $T^W \in \mathbb{R}^{N_w \times p}$ for extracting day-of-the-week characteristics. In this formulation, the parameter N_d indicates the total count of observational time slots per day, whereas $N_w = 7$ corresponds to the length of a standard week. Furthermore, to retrieve the appropriate contextual representations for any given moment t, we formulate a pair of mapping operations, $d(t)$ and $w(t)$. These indexing functions are responsible for identifying the exact position of the required embeddings within the respective memory pools. Utilizing these specific pointers, the model successfully extracts the corresponding intra-day component $T^{D}_{d(t)}$ alongside the weekly counterpart $T^{W}_{w(t)}$ for the active time step.

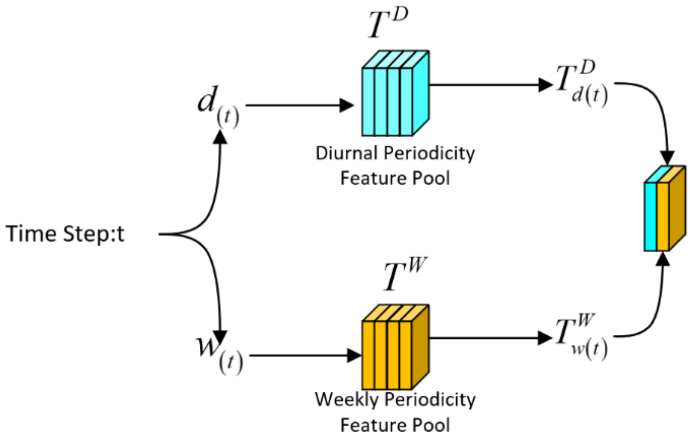

Fig. 2. The process of acquiring periodic features. This module utilizes a diurnal and weekly periodicity feature pool to generate time-specific embeddings, which are subsequently fused with real-time dynamic signals.

For traffic flow, the dynamic signal at a specific time step t can be expressed as:

$$F_t = MLP(X_t) \tag{1}$$

Where X_t represents the traffic flow data at time step t, and F_t denotes the dynamic signal extracted from X_t.

In real-world traffic systems, nodal traffic flow exhibits bidirectional interaction characteristics, manifested as dynamic processes of inflow and outflow. In this paper, we characterize these two types of dynamic signals as F_t^{in} and F_t^{out} respectively. While these signals reflect real-time interactions, they are often insufficient for identifying similar patterns across disparate time periods or predicting trends governed by periodic regularities. We therefore propose a dynamic-periodic coupling embedding mechanism that integrates dynamic signals with periodic features. This yields a graph representation with enhanced temporal generalization, defined as:

$$E_t^{in} = \tanh(F_t^{in} \odot T^{D}_{d(t)} \odot T^{W}_{w(t)}) \tag{2}$$

$$E_t^{out} = \tanh(F_t^{out} \odot T^{D}_{d(t)} \odot T^{W}_{w(t)}) \tag{3}$$

The periodic embedding vectors $T^{D}_{d(t)}$ and $T^{W}_{w(t)}$ encapsulate prior information regarding traffic intensity at the current time step. We fuse these with the real-time dynamic signal via an element-wise product, a mechanism designed to balance sensitivity with stability. When both the real-time signal and the periodic feature T exhibit high values in a specific dimension, this operation amplifies the feature, increasing its influence on the dynamic graph construction. Conversely, if the real-time signal undergoes an abrupt deviation—such as an isolated spike—while the periodic prior indicates historically low traffic, the product effectively dampens this fluctuation. This approach prevents the model from overreacting to transient noise, thereby enhancing robustness and ensuring adherence to inherent periodic regularities.

Subsequently, we construct the dynamic periodic graph by computing node similarity based on these fused representations. The formulation is defined as follows:

$$A_t^{in} = \mathrm{ReLU}(E_t^{in} E_t^{outT} - E_t^{out} E_t^{inT}) \tag{4}$$

$$A_t^{out} = \mathrm{ReLU}(E_t^{out} E_t^{inT} - E_t^{in} E_t^{outT}) \tag{5}$$

The generated matrices A_t^{in}, A_t^{out} correspond to the periodic dynamic adjacency matrices for the inflow and outflow directions at time step t. Here, we adopt an anti-symmetric structure of the form $E_t^{in} E_t^{out^T} - E_t^{out} E_t^{in^T}$ instead of the standard dot-product similarity. The rationale behind this design is to characterize the strong directionality and imbalance of traffic flow propagation. In the formulas, the cross-product terms ($E_t^{in} E_t^{out^T}$) represent the coupling strength between the two dynamic signals. Their subtraction essentially calculates the "net potential difference" of traffic interaction between nodes. By computing the difference between these two interaction strengths, the model can: (1) Automatically cancel out common background noise or weak symmetric correlations shared between nodes; (2) Retain only the significant dominant flow direction.

Consequently, the two generated adjacency matrices not only account for the influence of upstream nodes on downstream nodes but also explicitly consider the feedback effect of downstream nodes on upstream nodes.

4.1 Periodic Adaptive Graph Convolutional Recurrent Unit

Building on the utility of the Gated Recurrent Unit (GRU) in time-series forecasting, we introduce the Periodic Adaptive Graph Convolutional Recurrent Unit (PAGCRU). This enhanced architecture is engineered to jointly capture the spatio-temporal dynamics and periodic evolutionary regularities inherent in traffic flow.

Leveraging the Dynamic Graph Convolutional Network (AGCN) and the standard GRU, PAGCRU integrates periodic feature embeddings through a bidirectional dynamic graph interaction mechanism. This design enables the joint modeling of recurrent traffic patterns and dynamic spatial dependencies.

Real-world traffic interactions are inherently dynamic; for instance, congestion propagation patterns during morning rush hours diverge significantly from those observed during off-peak periods. Traditional static graph convolutions are often insufficient for modeling such time-varying topological correlations. To address this, we propose the Periodic Adaptive Graph Convolution Module, which utilizes the bidirectional adjacency matrices generated by the PDGM to perform adaptive feature extraction.

This design integrates periodic regularities while maintaining responsiveness to real-time states. Furthermore, to capture multi-scale spatial dependencies, we employ Chebyshev polynomial approximation for the Graph Laplacian. The polynomial order K defines the receptive field, allowing the model to aggregate information from neighbors up to K-1 hops away within the road network.

In traffic forecasting, spatial dependencies typically exhibit strong locality, meaning the mutual influence between road segments decays rapidly as distance increases. To balance the model's receptive field with computational complexity, while accounting for these physical propagation characteristics, we set the Chebyshev polynomial order to K=3. This configuration is empirically validated in the experimental section. The specific formulation is defined as follows:

$$AGCN(X,\{A_{in}^{t},A_{out}^{t}\})=\sum_{k=0}^{K-1}\sum_{s\in\{in,out,meta\}}W_{k}^{s}(A_{t}^{s})^{k}X \tag{6}$$

Where $X\in R^{B\times N\times D}$ represents the input features (where B denotes the batch size, N is the number of nodes, and D is the feature dimension). K denotes the order of the Chebyshev polynomials. $(A_{t}^{s})^{k}$ represents the k-th order adjacency matrix. W_{k}^{s} is the learnable weight matrix, which is utilized to fuse the features from k-th order neighbors.

Specifically, PAGCRU integrates dynamic graph convolution into the gating mechanism of the standard GRU , enabling the joint update of spatio-temporal features. The core formulations are defined as follows:

$$\begin{cases} z_{r}=\sigma(AGCN([X,H_{t-1}],E)) \\ z,r=split(z_{r}) \\ H_{t}=\tanh(AGCN([X,z\odot H_{t-1}]),E) \end{cases} \tag{7}$$

$$H_{t}=r\odot H_{t-1}+(1-r)\odot H_{t} \tag{8}$$

Where H_{t-1} denotes the hidden state at the previous time step, serving to store historical features. The notation $[X,H_{t-1}]$ denotes the feature concatenation of the current input and the previous hidden state. The symbol σ represents the Sigmoid activation function, which maps outputs to the $[0,1]$ interval to regulate gating signals. Specifically, z_{r} serves as a joint gating tensor encapsulating the raw outputs for both mechanisms; this vector is subsequently split along the feature dimension to decouple the independent update gate z and reset gate r. Furthermore, H_{t} represents the candidate hidden state, fusing the current input with the filtered historical state. H_{t} denotes the final hidden state at the current time step, integrating historical information with the latest spatio-temporal features.

4.2 Spatio-Temporal Meta-Graph Learner for Traffic Scenarios

Traffic flow data is characterized by strong periodicity, multi-modality, and significant spatio-temporal heterogeneity. Consequently, a single static graph structure is inadequate for characterizing the complex dependencies that vary across periodic stages and road segment types. To address this, MegaCRN[13] incorporating memory and discrimination capabilities into graph learning. However, their design remains relatively generic, lacking specific mechanisms to explicitly model the inherent periodic regularities, temporal coherence, and pattern stability unique to traffic dynamics.

To overcome the adaptability limitations in existing traffic forecasting models, we propose the Spatio-Temporal Meta-Graph Learner (STMG), explicitly tailored for traffic scenarios. By establishing a traffic pattern-aware meta-node bank and leveraging an enhanced query mechanism—which fuses periodic priors with road segment functional attributes—STMG generates reusable meta-graph topologies. This mechanism enables the precise retrieval of historical patterns, serving as a critical complement to the real-time dynamic graph structure.

To effectively characterize typical traffic patterns across diverse periodic stages, we design a Meta-Node Bank that encapsulates the prior features of traffic scenarios:

$$M = \{m_1, m_2, ..., m_N\}, m_i \in \mathrm{R}^d \tag{9}$$

Here, N denotes the total number of memory items. Each meta-node represents a reusable prototype of a spatio-temporal traffic pattern, and d corresponds to the feature dimension of each meta-node.

In constructing the meta-nodes, we explicitly integrate daily and weekly periodic traffic priors. This strategy ensures that the meta-nodes encapsulate pattern prototypes endowed with specific traffic semantics, distinguishing them from mere artifacts of simple feature clustering. The initialization is formalized as follows:

$$m_i = MLP_{init}(p_i^{day}, p_i^{week}) \tag{10}$$

$h_t \in \mathrm{R}^{N \times d_h}$ denote the hidden state output by the encoder. To ensure the query effectively captures variations in traffic scenarios, we incorporate periodic embeddings into the construction of the query vector. Subsequently, we compute the similarity between this query vector and the Meta-Node Bank. A Softmax function is then applied to derive attention weights, which are utilized to generate a composite meta-node representation tailored to the current traffic scenario.

$$q_t = W_q[h_t \,\|\, e_t^{day} \,\|\, e_t^{week}] \tag{11}$$

$$\begin{cases} s_i = \langle q_t, m_i \rangle \\ \alpha_i = \dfrac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)} \\ m_t = \sum_{i=1}^{N} \alpha_i m_i \end{cases} \tag{12}$$

In the formulation, e_t^{day} and e_t^{week} represent the periodic embeddings, while W_q denotes a learnable weight matrix. The symbol $\|$ indicates the concatenation operation. s_i represents the similarity score, α_i denotes the attention weight, and m_t signifies the typical traffic pattern that best aligns with the current periodic stage.

The reconstructed meta-node vector m_t is used to augment the encoded hidden representation h_t, yielding the augmented state $h_t^{'} = [h_t, m_t]$. To transform this pattern vector into a specific graph structure, we employ a Hyper-network Φ_θ to generate the meta-graph adjacency matrix A_t^{meta}. Specifically, the Hyper-network first generates the node embedding $E^{meta} \in \mathrm{R}^{N \times \mathrm{d_e}}$, and subsequently constructs the adjacency matrix via non-negative embedding similarity. This meta-graph effectively embodies the long-term spatial pattern structure likely to emerge in the traffic system under the current periodic scenario.

$$\begin{cases} E_t^{meta} = \Phi_\theta(h_t^{'}) \\ A_t^{meta} = \mathrm{Softmax}(\mathrm{ReLU}(E_t^{meta} (E_t^{meta})^{\mathrm{T}})) \end{cases} \tag{13}$$

The generated meta-graph A_t^{meta} serves as a substitute for the periodic dynamic graph and is feedback to the Graph Convolutional Recurrent Unit (GCRU), providing dynamic topological support for subsequent spatio-temporal modeling. The specific structure of this module is illustrated in Fig. 3.

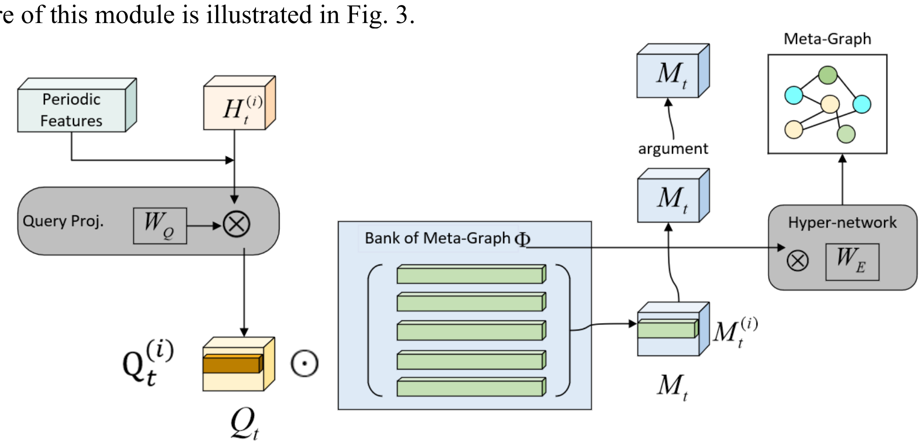

Fig. 3. Spatio-Temporal Meta-Graph Learner Module Structure. It details the attention-based query process matching current periodic features with the meta-node bank, and the subsequent hyper-network generating the context-specific meta-graph.

4.3 Meta-Graph Convolutional Recurrent Unit

Building upon the Meta-Graph Learner described in the preceding section, this paper further introduces the Meta-Graph Convolutional Recurrent Unit. The primary objective of this unit is to learn discriminative and representative pattern prototypes within the meta-node bank, thereby enabling the adaptive update of the auxiliary graph in response to observed traffic conditions.

To enhance the model's generalization capabilities, we optimize the memory parameters by enforcing two constraint mechanisms: Contrastive Loss and Consistency Loss.

(1) Contrastive Loss (L_1): Leveraging the principles of triplet loss, this component enforces the distance between the local query vector $Q_t^{(i)}$ and its most similar pattern ϕ_p to be smaller than the distance to the second most similar pattern ϕ_n by at least a predefined margin λ . This mechanism establishes a robust discriminative boundary, explicitly guiding the memory bank to learn distinct and well-separated pattern prototypes. Consequently, this design prevents redundancy and overlap among stored patterns, thereby enhancing the model's capacity to capture spatio-temporal heterogeneity. The formulation is defined as:

$$L_1 = \sum_{t,i}^{T,N} \max(\| Q_t^{(i)} - \phi[p] \|^2 - \| Q_t^{(i)} - \phi[n] \|^2 + \lambda, 0) \tag{14}$$

(2) Consistency Loss(L_2): This loss ensures that the Euclidean distance between the local query vector $Q_t^{(i)}$ and its most correlated memory item ϕ_p is minimized. This constraint facilitates the precise mapping of the current traffic scenario onto its best-matching historical pattern prototype, achieving precise pattern reuse. The formulation is defined as:

$$L_2 = \sum_{t,i}^{T,N} \| Q_t^{(i)} - \phi[p] \|^2 \tag{15}$$

In the above equations, T represents the sequence length in the training set. p and n denote the indices of the top-two memory items, obtained by sorting the similarity weights with respect to the local query vector $Q_t^{(i)}$. Specifically, p corresponds to the index of the most similar memory item, while n corresponds to the index of the second most similar memory item.

The selection criteria for the most similar memory item p and the second most similar memory item n strictly rely on the ranking of Euclidean distances between the local query vector $Q_t^{(i)}$ and all memory items within the meta-node bank ϕ .During the training process of the Meta-Graph Learner, the optimization of the contrastive loss L_2 effectively clusters query vectors sharing similar traffic semantic features and spatio-temporal evolution patterns around the same meta-node ϕ_j .Consequently, the Euclidean distance in this mechanism serves as a semantic distance metric between pattern prototypes. The specific role of L_1 is to compel the model to learn patterns that are not closely adjacent in the feature space, thereby preventing pattern redundancy and overlap. This constraint guides the memory bank ϕ to differentiate spatio-temporal patterns at the node level, ensuring that the stored items represent distinct traffic prototypes. This effectively improves the model's ability to capture spatio-temporal heterogeneity.

5 Experiments

5.1 Dataset Description And Analysis

The forecasting capabilities of the proposed PDMetaNet are verified using two widely adopted, real-world benchmark datasets: PeMS04 (comprising 307 sensor nodes and 16,992 temporal points) and PeMS08 (containing 170 nodes and 17,856 temporal points). These traffic records are sourced from the Caltrans Performance Measurement System. The raw data is sampled at a 5-minute frequency, resulting in 288 discrete data frames for each 24-hour cycle[14][15][16].

For data preprocessing, the raw sensor measurements undergo Z-score scaling to eliminate dimensional variations, effectively speeding up the network training phase. To avoid future information leakage, the entire timeline is strictly divided into training, validation, and testing segments according to a 7:1:2 proportion. Prior to formal modeling, a preliminary temporal analysis of PeMS04 revealed a rigorous bimodal distribution during workdays, along with a high correlation coefficient (exceeding 0.89) between successive weekends. This observation strongly justifies our motivation to incorporate cyclic feature embeddings. A visual representation of the traffic volume over three consecutive weekdays (January 15 to 17, 2018) alongside three sequential Saturdays is depicted in Fig. 4.

Regarding the computational environment, the model implementation relies on Python 3.8 and the PyTorch 1.10.0 library, executed on a workstation featuring a single NVIDIA RTX 4090 graphics card (24GB memory) coupled with an Intel Xeon Gold 6430 processor. For the sequence configurations, the length of the historical observation window matches the target prediction span, both fixed at 12 steps (equivalent to one hour). Network parameters are updated using the Adam algorithm over a maximum of 200 epochs, with the batch capacity constrained to 64 samples. The starting learning rate is established at 0.01, and it experiences a 10-fold reduction at the 50th and 100th epochs. To avert the risk of overfitting, the optimization process halts prematurely if the validation performance shows no improvement for 20 consecutive rounds. Ultimately, the network minimizes the Masked Mean Absolute Error (Masked MAE) objective function, while a curriculum learning strategy is activated to guarantee smooth convergence.

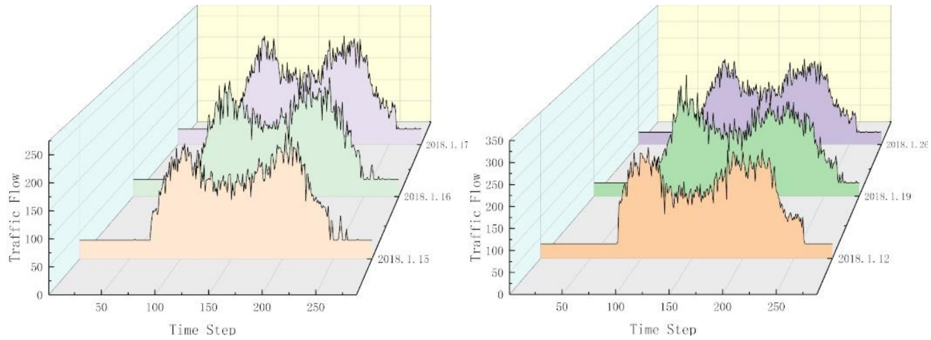

Fig. 4. Three-Day Traffic Flow of the PEMS04 Dataset and Three-Consecutive-Saturday Traffic Flow of the PEMS04 Dataset

We evaluate the prediction performance of the proposed model using three standard metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE)[14],[17]. These are defined as follows:

$$E_{\mathrm{MAE}} = \frac{1}{N}\sum_{i=0}^{N} | Y_i - Y_i | \tag{16}$$

$$E_{\mathrm{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=0}^{N} (Y_i - Y_i)^2} \tag{17}$$

$$E_{\mathrm{MAPE}} = \frac{1}{N}\sum_{i=0}^{N} \frac{| Y_i - Y_i |}{Y_i} \tag{18}$$

5.2 Experimental Result

To comprehensively evaluate our approach, we compare PDMetaNet against ten representative baselines, ranging from classical sequence models to state-of-the-art spatio-temporal graph neural networks. These include the Temporal Convolutional Network (TCN[17])， static graph-based models (STGCN[5], ASTGCN[6], STSGCN[18]), dynamic and adaptive graph networks (AGCRN[7], DSTAGNN[14]）and other recent advanced architectures (STFGCN[19], STGODE[20], STIIPGCN[15],TDDCG[16]).

Table 1 and Figures 5-6 detail the prediction performance of all models on the PeMS04 and PeMS08 datasets. PDMetaNet consistently outperforms all baselines across all metrics on both datasets. On PeMS04, it improves upon the highly competitive STIIPGCN model, reducing MAE, RMSE, and MAPE by 5.4%, 2.1%, and 3.7%, respectively. On PeMS08, PDMetaNet lowers MAE and RMSE by 3.1% and 5.1% compared to the next best method, while achieving a comparable MAPE.

Furthermore, compared to traditional models relying on static graph structures (e.g., STGCN, ASTGCN), PDMetaNet dynamically adjusts nodal association weights in real-time, resulting in an average MAE reduction of over 15%. This suggests that our periodic dynamic graph mechanism adapts more effectively to spatial topological changes. Additionally, in contrast to models lacking periodic memory mechanisms (e.g., AGCRN), the meta-graph memory module in PDMetaNet explicitly extracts and reuses cross-period patterns. This directly contributes to more stable performance in long-sequence forecasting, achieving an average MAPE reduction of 5%–8%. The demonstrated superiority, particularly in the RMSE metric, further indicates that PDMetaNet possesses stronger robustness against abnormal fluctuations in complex traffic networks.

Table 1. Prediction Performance Comparison

Models	PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
TCN	23.22	37.26	15.59%	22.72	35.79	14.03%
STGCN	21.16	34.89	13.83%	17.50	27.09	11.29%
ASTGCN	22.93	35.22	16.56%	18.25	28.06	11.64%
STSGCN	21.19	33.65	13.90%	17.13	26.80	10.96%
AGCRN	19.83	32.26	12.97%	15.95	25.22	10.09%
STFGNN	20.48	32.51	16.70%	16.94	26.25	10.60%
STGODE	20.84	32.82	13.77%	16.81	25.97	10.62%
DSTAGNN	19.30	31.46	12.70%	15.67	24.77	9.94%
STIIPGCN	19.32	30.77	12.75%	14.66	23.55	**9.59%**
TDDCG	21.52	34.29	15.29%	15.66	25.36	10.80%
PDMetaNet	**18.27**	**30.12**	**12.28%**	**14.20**	**22.35**	9.61%

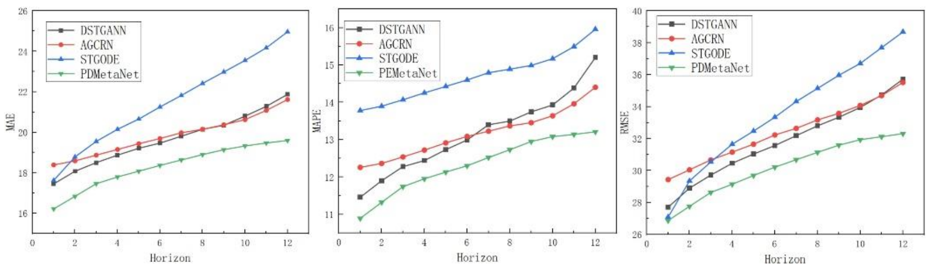

Fig. 5. Prediction Performance Comparison Between PDMetaNet and State-of-the-Art Models on the PEMS04 Dataset

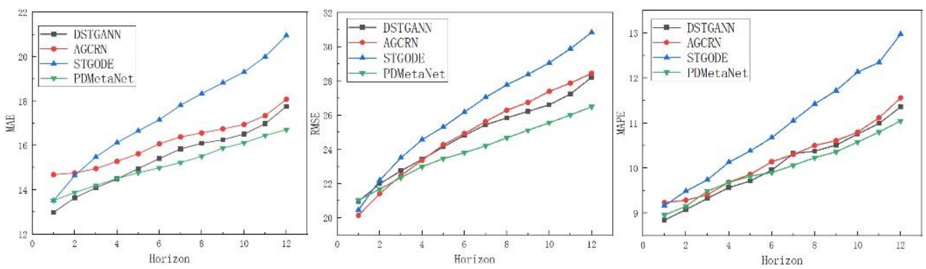

Fig. 6. Prediction Performance Comparison Between PDMetaNet and State-of-the-Art Models on the PEMS08 Dataset

5.3 Ablation Study

To verify the effectiveness of each specific module within PDMetaNet, this section conducts comparative experiments between PDMetaNet and its variant models. The defined variants are as follows:

(1) w/o Pe: Removes periodic embeddings; the dynamic graph is constructed solely based on dynamic signals.

(2) w/o PDGM: Removes the Periodic Dynamic Graph Generation Module and replaces it with a static graph.

(3) w/o PDGCN: Removes the Periodic Dynamic Graph Convolution and substitutes it with a standard GCN.

(4) w/o Enh: The Spatio-Temporal Meta-Graph Learner constructs the meta-graph but excludes information enhancement.

(5) w/o Mega: The Spatio-Temporal Meta-Graph Learner performs information enhancement but does not construct the meta-graph; the decoder utilizes the graph generated by PDGM instead.

(6) w/o STG: Completely removes the Spatio-Temporal Meta-Graph Learner (neither enhancement nor meta-graph construction is performed), and the decoder uses the graph from PDGM.

Based on the performance comparison between PDMetaNet and its six variants on the PEMS04 and PEMS08 datasets (presented in Table 2), the contribution and synergistic value of each core module to prediction accuracy are clarified below:

The removal of any module related to the periodic dynamic graph leads to a significant degradation in performance. The w/o PDGM variant, which replaces the dynamic graph generation module with a static graph, shows the most significant performance decline. On the PEMS04 and PEMS08 datasets, MAE increases by 24.4% and 22.7%, while MAPE rises by 34.5% and 41.2%, respectively. This indicates that the dynamic graphs generated by PDGM are essential for adapting to real-time changes in spatial correlations. Additionally, the w/o Pe variant results in an average MAE increase of 9.5%, confirming that periodic patterns offer important priors for dynamic graph construction. Finally, the w/o PDGCN variant, which uses a standard GCN, leads to an average MAE increase of 6.1%, highlighting the benefit of dynamic convolution for multi-order spatial feature propagation.

The hierarchical ablation study confirms the critical role of the STG module. Completely removing this module results in an average MAE increase of 17.2% and a MAPE increase of 23.7%, highlighting the importance of reusing long-term historical patterns. We also examined the specific contributions of its components: removing only the enhancement component leads to an average MAE increase of 3.9%, whereas removing the meta-graph construction increases MAE by 7.6%. These results demonstrate that combining meta-graph construction with information enhancement effectively improves the utilization of historical patterns.

The full PDMetaNet model outperforms all variants, with the most significant performance drops occurring when the PDGM and STG modules are removed. This suggests that the model's effectiveness relies on the synergy between capturing time-varying spatio-temporal dependencies and reusing historical patterns. Specifically, the dynamic graph provides time-varying inputs to the meta-graph, while the meta-graph supplies historical priors. This interaction ensures both real-time adaptability and long-term robustness, resulting in more precise predictions.

Table 2. Ablation Comparison Experimental Results

Variant Models	PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
PDMetaNet	**18.27**	**30.12**	**12.28%**	**14.20**	**22.35**	**9.61%**
w/o Pe	19.85	32.67	13.75%	15.68	24.52	10.83%
w/o PDGM	22.73	36.15	16.52%	17.42	28.88	13.57%
w/o PDGCN	19.32	31.89	13.16%	15.14	23.79	10.25%
w/o Enh	18.96	31.43	12.89%	14.77	23.21	9.94%
w/o Mega	19.53	32.28	13.41%	15.36	24.15	10.52%
w/o STG	21.08	34.86	15.03%	16.89	26.53	12.01%

5.4 Parameter Sensitivity Analysis

We investigated the impact of the Chebyshev polynomial order K in the Periodic Adaptive Graph Convolution through sensitivity experiments on the PEMS04 and PEMS08 datasets. We tested four values $\mathrm{K} = \{2,3,4,5\}$, and the results are presented in Table 3.

PDMetaNet achieves optimal performance across all metrics when $\mathrm{K} = 3$. As K increases further, performance shows a slight but consistent decline. This occurs because a larger K expands the receptive field, introducing noise from high-order neighbors and exacerbating over-smoothing. generally, high-order neighbors have lower correlations with short-term traffic flow, so fusing their features can dilute essential local dependency information. Therefore, setting $\mathrm{K} = 3$ allows the model to capture key local spatial dependencies efficiently while keeping computational costs low.

Table 3. Parameter Sensitivity Analysis Results

K value	PEMS04			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
2	18.51	30.55	12.50%	14.45	22.80	9.96%
3	**18.27**	**30.12**	**12.28%**	**14.20**	**22.35**	**9.61%**
4	18.80	30.93	12.75%	14.75	23.30	10.27%
5	19.12	31.40	13.07%	15.10	23.92	10.60%

5.5 Discussion

The consistent superiority of PDMetaNet, particularly its enhanced robustness in complex traffic networks, stems from the synergistic interplay between the PDGM and

STMG modules. PDGM endows the model with the necessary agility to respond to real-time traffic fluctuations and abrupt state deviations, such as temporary bottlenecks caused by unexpected accidents. Simultaneously, the STMG module serves as an empirical knowledge base, retrieving prototypical peak-hour patterns to prevent the dynamic graph from topological collapse when faced with extreme sensor noise or transient outliers. This mutual constraint between long-term periodic priors and short-term dynamic adaptability is the primary reason for the substantial reduction in the RMSE metric, which reflects the model's resilience against abnormal fluctuations.

In terms of model complexity, the adoption of Chebyshev polynomial approximation with an order of K=3 enables the aggregation of spatial information from local neighbors without incurring the $O(L^2)$ computational cost of global matrix operations. This ensures that PDMetaNet remains computationally efficient and practically applicable to large-scale networks. Furthermore, to manage the parameter uncertainty introduced by the initialization and attention-based querying of the meta-node bank, we incorporate Contrastive and Consistency Losses. These constraints effectively regularize the parameter space, preventing pattern redundancy and significantly enhancing the stability and generalization of the memory mechanism.

6 Conclusion

To address the challenges of dynamic spatio-temporal dependencies, periodic pattern reuse, and spatio-temporal heterogeneity in traffic flow prediction, we propose PDMetaNet. The model features a three-level framework consisting of Periodic Dynamic Graph Generation, Periodic Adaptive Graph Convolution Modeling, and Spatio-Temporal Meta-Graph Memory Enhancement. The Periodic Dynamic Graph Generation Module (PDGM) produces bidirectional time-varying adjacency matrices by coupling dynamic signals with periodic feature embeddings. This approach allows the model to adapt to real-time changes in road network spatial correlations, overcoming the limitations of static graphs. Meanwhile, the Periodic Adaptive Graph Convolution Recurrent Unit (PAGCRU) integrates dynamic graph convolution with gated recurrent mechanisms to simultaneously extract and update spatio-temporal features. Finally, the Spatio-Temporal Meta-Graph Learner stores cross-period patterns in a meta-node bank and generates meta-graphs by combining attention queries with hyper-networks. This mechanism addresses the challenge of reusing long-term historical patterns, combining dynamic modeling with historical reuse.

Comparative evaluations on the PEMS04 and PEMS08 datasets show that PDMetaNet consistently outperforms ten mainstream baseline models. This confirms the model's adaptability to varying network scales and its high prediction accuracy. In practice, PDMetaNet removes the need for predefined road network topologies, adaptively capturing traffic periodicity, dynamics, and spatio-temporal heterogeneity. It maintains stable performance in both short-term (15–30 minutes) and long-term (60 minutes) forecasting. The model is particularly effective in complex scenarios, such as modeling congestion propagation during peak hours and capturing dynamic inter-link correlations. Consequently, it provides precise traffic state prediction support for signal optimization, route planning, and emergency response within Intelligent Transportation Systems (ITS).

References

1. Li, Y., et al.: A systematic review of generative adversarial networks for traffic state prediction: Overview, taxonomy, and future prospects. Inf. Fusion 115, 102915 (2025)
2. Yao, J., et al.: Review of traffic flow prediction methods based on machine learning. J. Traffic Transp. Eng. 23(3), 44–67 (in Chinese) (2023)
3. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: Proc. Int. Conf. Learn. Represent. (ICLR). Vancouver, BC, Canada (2018)
4. Xu, Y., et al.: Dynamic traffic correlations based spatio-temporal graph convolutional network for urban traffic prediction. Inf. Sci. 621, 580–595 (2023)
5. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI), pp. 3634–3640. Stockholm, Sweden (2018)
6. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proc. 33rd AAAI Conf. Artif. Intell., pp. 922–929. Honolulu, HI, USA (2019)
7. Bai, L., et al.: Adaptive graph convolutional recurrent network for traffic forecasting. In: Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 17804–17815. Vancouver, BC, Canada (2020)
8. Pareja, A., et al.: EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In: Proc. AAAI Conf. Artif. Intell. 34(4), pp. 5363–5370 (2020)
9. Zhao, W., et al.: Periodicity aware spatial-temporal adaptive hypergraph neural network for traffic forecasting. GeoInformatica 29(2), 201–232 (2025)
10. Fan, J., et al.: Pdg2seq: Periodic dynamic graph to sequence model for traffic flow prediction. Neural Netw. 183, 106941 (2025)
11. Zhang, H., et al.: Two-way heterogeneity model for dynamic spatiotemporal traffic flow prediction. Knowl.-Based Syst., 113635 (2025)
12. Yang, S., Wu, Q., Li, M.: Decoupled multi-spatio-temporal fusion graph convolutional recurrent network for traffic prediction. Eng. Appl. Artif. Intell. 163, 112956 (2026)
13. Jiang, R., et al.: Spatio-temporal meta-graph learning for traffic forecasting. In: Proc. AAAI Conf. Artif. Intell. 37(7), pp. 8078–8086 (2023)
14. Lan, S., et al.: DSTAGNN: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In: Proc. Int. Conf. Mach. Learn. (ICML), pp. 11906–11917. Baltimore, MD, USA (2022)
15. Zhang, H., et al.: Traffic flow prediction method based on key graph and spatiotemporal information interaction. Comput. Eng. Appl. (in Chinese) (2025)
16. Yang, G., et al.: Short-term traffic flow spatiotemporal prediction based on transformer. Comput. Appl. Softw. 41(3), 169–173 (in Chinese) (2024)
17. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
18. Weng, W.: Traffic prediction based on spatiotemporal graph neural network. M.S. thesis, Hangzhou Dianzi Univ., Hangzhou, China (in Chinese) (2024)

19. Li, M., Zhu, Z.: Spatial-temporal fusion graph neural networks for traffic flow forecasting. In: Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD), pp. 4189–4198. Singapore (2021)
20. Fang, Z., et al.: Spatial-temporal graph ODE networks for traffic flow forecasting. In: Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining (KDD), pp. 364–373. New York, NY, USA (2021)