Contents lists available at ScienceDirect

# Information Fusion

Full length article

# COOL: A Conjoint Perspective on Spatio-Temporal Graph Neural Network for Traffic Forecasting

Wei Ju [a,1], Yusheng Zhao [a,1], Yifang Qin [a], Siyu Yi [b], Jingyang Yuan [a], Zhiping Xiao [c], Xiao Luo [c,*], Xiting Yan [d,e], Ming Zhang [a,*]

[a] *School of Computer Science, National Key Laboratory for Multimedia Information Processing, Peking University, Beijing, 100871, China*
[b] *School of Statistics and Data Science, Nankai University, Tianjin, 300071, China*
[c] *Department of Computer Science, University of California, Los Angeles, 90095, USA*
[d] *Section of Pulmonary, Critical Care and Sleep Medicine, Yale School of Medicine, New Haven, CT, 06520, USA*
[e] *Department of Biostatistics, Yale School of Public Health, New Haven, CT, 06520, USA*

## ARTICLE INFO

## ABSTRACT

This paper investigates traffic forecasting, which attempts to forecast the future state of traffic based on historical situations. This problem has received ever-increasing attention in various scenarios and facilitated the development of numerous downstream applications such as urban planning and transportation management. However, the efficacy of existing methods remains sub-optimal due to their tendency to model temporal and spatial relationships independently, thereby inadequately accounting for complex high-order interactions of both worlds. Moreover, the diversity of transitional patterns in traffic forecasting makes them challenging to capture for existing approaches, warranting a deeper exploration of their diversity. Toward this end, this paper proposes Conjoint Spatio-Temporal graph neural network (abbreviated as COOL), which models heterogeneous graphs from prior and posterior information to conjointly capture high-order spatio-temporal relationships. On the one hand, heterogeneous graphs connecting sequential observation are constructed to extract composite spatio-temporal relationships via prior message passing. On the other hand, we model dynamic relationships using constructed affinity and penalty graphs, which guide posterior message passing to incorporate complementary semantic information into node representations. Moreover, to capture diverse transitional properties to enhance traffic forecasting, we propose a conjoint self-attention decoder that models diverse temporal patterns from both multi-rank and multi-scale views. Experimental results on four popular benchmark datasets demonstrate that our proposed COOL provides state-of-the-art performance compared with the competitive baselines.

## 1. Introduction

Spatio–temporal forecasting [1,2] has emerged as a prominent research area due to its relevance in numerous downstream applications. From urban planning and environmental management to logistics optimization and beyond, accurate predictions of how entities evolve over time and space are essential. One particularly critical real-world problem in this domain is traffic flow forecasting [3–6], which aims to forecast future traffic based on historical situations. It involves predicting various aspects of traffic dynamics, including traffic volume, speed, and congestion patterns, across different locations and time intervals. The applications of traffic flow prediction are far-reaching, impacting intelligent transportation systems [3], traffic management [7],

route planning [8], and ultimately contributing to reduced congestion, improved transportation efficiency, and enhanced urban livability.

Recently, a range of algorithms for effective traffic flow prediction have been proposed, broadly categorized as physics-based and learning-based. For the former, physics-based methods typically rely on differential equations to formally describe traffic systems [9,10]. They often exhibit outstanding performance in simulated environments, supported by rigorous theoretical foundations. However, these methods often struggle to adapt to the complexities of real-world scenarios due to their demanding model assumptions [11]. Conversely, for the latter, learning-based approaches are widely adopted for their ability to optimize machine learning models using historical observations, making them a popular choice for predicting future trends. Early research

---

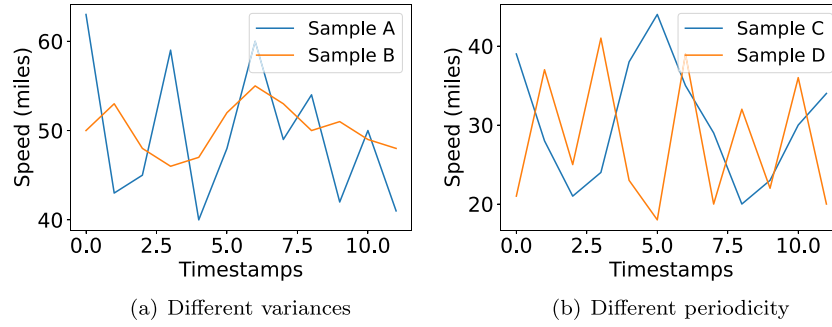(a) Different variances        (b) Different periodicity

**Fig. 1.** Transitional patterns in traffic networks could be diverse, varying in terms of variances (a) and periodicity (b).

endeavors attempt to tackle this challenge with traditional models such as Autoregressive Integrated Moving Average [12] and Support Vector Machines [13]. However, their modeling capacity is insufficient to fit large-scale and complex data. In recent years, deep learning-based approaches have gained considerable attention due to their leverage of the powerful representation learning capabilities of deep neural networks, resulting in significant improvements. On the one hand, they employ Recurrent Neural Networks [14] or Temporal Convolutional Networks [15] to capture temporal dependencies in the traffic data. TE-TCN [16] proposes Transformer-enhanced temporal convolutional networks to capture long- and short-term periodic dependencies. On the other hand, Graph Neural Networks (GNNs) are utilized to extract structured spatial relationships from road networks. DyHSL [6] leverages hypergraph structural information for non-pairwise spatial relationships. By harnessing the strengths of both worlds, current algorithms can effectively capture temporal and spatial information, facilitating accurate traffic flow predictions [5,17–20]. For example, GDGCN [5] develops a novel temporal graph convolutional block for flexible temporal relations, and a dynamic graph constructor to model both time-specific spatial dependencies and changing temporal interactions. DS-TGCN [20] incorporates spatial–temporal similarity feature and convolution with an attention mechanism to effectively extract complex spatial–temporal relationships.

Nevertheless, despite the impressive performance achieved by existing traffic flow forecasting techniques [18,21–24], they still suffer from two critical flaws: (1) *Fail to effectively capture composite spatio–temporal relationships.* For example, consider the failure to adequately model the interplay between traffic congestion and local weather conditions, such as rain or snow, which can significantly impact traffic flow dynamics. Existing spatio–temporal algorithms typically combine GNNs and RNNs by fusing the corresponding representations in traffic networks [25–27]. Unfortunately, this naive combination separates the mining of spatial and temporal correlations. In detail, these methods [28,29] cannot acquire various temporal information while extracting spatial messages, thus missing high-order composite relationships. Worse still, they usually extract spatial correlations from road networks, which neglects dynamic semantic correlations in traffic systems, resulting in inferior performance for traffic prediction. (2) *Failing to sufficiently capture diverse transitional patterns.* Due to diverse traffic requirements, different locations or times could exhibit various transitional patterns. For example, existing models may have difficulty accurately predicting transitions between regular weekday traffic flow and the highly variable traffic patterns during holidays or special events like concerts or sports games. Furthermore, they might not effectively capture the transition patterns between workdays and weekends, which often exhibit distinct traffic dynamics due to changes in commuter behavior. As shown in Fig. 1, locations could show different periodic patterns due to daily or weekly routines [17]. However, existing methods mostly fail to model complex temporal dependencies effectively using standard sequential models, hindering them from making accurate traffic low predictions.

In this paper, we present a novel approach named COOL for effective traffic prediction. At a high level, COOL conjointly explores high-order spatio–temporal relationships from both prior and posterior information. On the one hand, we introduce prior information into constructed heterogeneous graphs connected by spatial and temporal connections. On the other hand, we model dynamic relationships using both constructed affinity and penalty graphs, then a posterior message passing layer is developed to incorporate both similarities and dissimilarities into sequential node representations. In addition, to capture diverse transitional properties to enhance traffic forecasting, we develop a conjoint self-attention decoder that aggregates sequential representations by modeling diverse temporal patterns from both multi-rank and multi-scale views. In particular, we not only utilize transformation matrices of different sizes to provide multi-rank attention matrices to model diverse intrinsic patterns, but also involve multi-scale pooling to generate subsequence representations for capturing various periodic patterns. Eventually, we combine obtained global representations in an adaptive manner to generate traffic predictions.

Compared to the current state-of-the-art baseline method STAE-former [30], which primarily employs different Transformer layers to sequentially model temporal and spatial information, independently modeling them for two inherently coupled aspects often leads to sub-optimal performance. In contrast, our proposed COOL naturally couples temporal and spatial information by constructing a heterogeneous graph. Through both prior and posterior message passing, COOL further captures high-order spatio–temporal dependencies. Additionally, STAEformer does not consider the rich temporal patterns present in real-world scenarios of traffic flow prediction, while our COOL leverages the self-attention mechanism [31] to model these patterns and capture dynamics more realistically.

The contributions of this paper can be summarized as follows:

- We propose a novel spatio–temporal graph convolutional network model COOL, which conjointly explores high-order spatio–temporal relationships using both prior and posterior information for traffic flow prediction.
- To capture various long-term transitional patterns, COOL introduces a conjoint self-attention decoder that aggregates sequential representations using both multi-rank and multi-scale attention branches.
- Extensive experiments on four benchmark datasets achieve promising results and outperform the competitive baseline methods by a large margin, which validates the effectiveness of our method.

## 2. Related work

### 2.1. Graph neural networks

GNNs have gained significant popularity in recent years due to their effectiveness in modeling structured data with complex relationships [32–38]. The fundamental idea behind GNNs is to learn

representations of nodes in a graph by aggregating information from their neighbors, thereby capturing the inherent graph structure [39]. GNNs have been extensively applied to various downstream tasks, including node classification [40–42], graph classification [43–48], and graph clustering [49–51], where they have demonstrated impressive performance. In the context of spatio–temporal analysis, GNNs have been extensively applied. Several previous works in spatio–temporal analysis have utilized GNNs to model various phenomena [6,15,18, 52,53]. For example, Graph WaveNet [15] captures spatial–temporal dependencies using an adaptive dependency matrix learned through node embeddings, and handles long sequences efficiently with stacked dilated 1D convolution components. STSGCN [18] efficiently captures complex localized spatial–temporal correlations using a designed synchronous modeling mechanism and accounts for heterogeneities in localized spatial–temporal graphs through multiple time period modules. CNFGNN [52] proposes a federated spatio–temporal model that leverages GNN-based architecture to encode the graph structure, constrained by cross-node federated learning, which disentangles temporal and spatial dynamics while reducing communication costs. While previous spatio–temporal analysis methods based on GNNs have made notable contributions, our proposed approach COOL stands out by addressing critical limitations and providing superior performance in capturing complex spatio–temporal relationships and diverse transition patterns in traffic data, these innovations lead to more accurate and robust traffic flow predictions.

### 2.2. Traffic flow forecasting

Traffic Flow Forecasting is a well-recognized and highly extensively researched problem in the field, garnering significant attention and interest [2]. Numerous spatio–temporal forecasting methods have been adapted for this task, yielding remarkable results. The predominant approach in addressing this challenge is rooted in machine learning algorithms, which leverage spatio–temporal data collected from an array of sensors to predict future traffic conditions. Traditional methods such as k-nearest neighbors [29], autoregressive integrated moving average [12], and support vector machines [13] have been employed, however, they often fall short in effectively modeling complex spatial relationships inherent in traffic data. With the rapid advancements in deep neural networks, deep learning-based methods have emerged as the dominant paradigm, focusing on the intricate modeling of spatio–temporal dependencies within traffic flow data [54]. The fundamental idea revolves around utilizing deep neural architectures to capture these dependencies. This entails harnessing the power of GNNs to extract structured spatial relations encoded within road networks. Concurrently, sequence neural networks excel in capturing temporal dependencies over time. These two complementary approaches are often integrated to develop comprehensive models capable of handling the multifaceted intricacies of traffic flow forecasting [17,25–27]. For instance, ASTGCN [17] models recent, daily-periodic, and weekly-periodic traffic dependencies through spatial–temporal attention mechanisms, graph convolutions for spatial patterns, and standard convolutions for temporal features, with fused outputs for predictions. GMAN [25] introduces a graph multi-attention network that predicts future traffic conditions on a road network graph using an encoder–decoder architecture with spatio–temporal attention blocks. DGCRN [27] leverages hypernetworks to extract dynamic node attributes and generates dynamic filters at each time step for filtering node embeddings. However, existing spatio–temporal GNN methods still have some limitations in capturing high-order relationships and diverse transitional properties. To tackle this, we propose a novel method named COOL, which not only conjointly explores high-order spatio–temporal correlations in constructed heterogeneous graphs extracted from both prior and posterior information, but also incorporates a conjoint self-attention decoder that leverages both multi-rank and multi-scale self-attention to capture diverse temporal transitional patterns.

## 3. Methodology

This paper presents a new approach named COOL for traffic flow forecasting. As illustrated in Fig. 2, COOL consists of a conjoint spatio–temporal graph encoder and a conjoint self-attention decoder. In the encoder, we conjointly extract high-order spatio–temporal correlations from both prior and posterior information. On the one hand, we perform message passing under the guidance of constructed heterogeneous graphs containing both prior spatial and temporal connections. On the other hand, we construct both semantic affinity graphs and penalty graphs to characterize dynamic relationships, and then incorporate both similarities and dissimilarities into sequential node representations. In the decoder, we aggregate sequential representations by exploring diverse transitional patterns from both multi-rank and multi-scale views.

**Problem Definition.** We denote the traffic graph of road network as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the node set $\mathcal{V}$ and the edge set $\mathcal{E}$. The adjacency matrix can be written as $A \in \mathbb{R}^{N \times N}$. The historical observation can be denoted as $\{X_1, X_2, \ldots, X_T\}$ where $X_t \in \mathbb{R}^{N \times F}$ is the observation at $t$ time step, $F$ is the dimension of each observation. The key objective of traffic flow forecasting is to predict the future observations $X_t (t > T)$.

### 3.1. Conjoint spatio–temporal graph encoder

The encoder is comprised of two parts. It first constructs heterogeneous graphs for prior message passing and then infers affinity and penalty graphs for posterior message passing.

**Heterogeneous Graph Generator.** To learn composite spatio–temporal relationships, we present a heterogeneous graph that connects observations from both temporal and spatial views.

In detail, we consider $r$ time steps in each heterogeneous graph $\mathcal{G}^H_{[t-r+1:t]}$, containing $rN$ observations $\{v_i^t\}_{t \in [t-r+1:t], v_i \in \mathcal{V}}$. The two observations are connected using spatial edges, i.e., $w_{v_i^t, v_j^t} = A_{ij}$ at each time step while consecutive observations are also connected using temporal edges, i.e., $w_{v_i^t, v_i^{t+1}} = 1$.

**Prior Message Passing.** Then, we introduce prior message passing to capture composite correlations in traffic data. In particular, we leverage the message passing mechanism where each representation is updated by aggregating information from its neighborhood [39]. Formally, the representation of $v_i^t$ at the $k$th layer $h_i^{t,(k)}$ can be written as:

$$h_i^{t,(k)} = C_\theta^{(k)} \left( h_i^{t,(k-1)}, A_\theta^{(k)} \left( \left\{ h_j^{t',(k-1)} \right\}_{v_j^{t'} \in \mathcal{N}(v_i^t)} \right) \right), \tag{1}$$

where $\mathcal{N}(v_i^t)$ is the neighbors of $v_i^t$. $A_\theta^{(k)}$ and $C_\theta^{(k)}$ denote the aggregation and combination operations parameterized by $\theta$ at the $k$th layer, respectively. The final embedding of the node $v_i$ at time step $t$ and the $K$th layer can be written as $h_i^t = h_i^{t,(K)}$.

**Affinity and Penalty Graph Generator.** Nevertheless, prior message passing merely considers the affinities between pairs of nodes and often assigns zero weight to characterize dissimilarity [18,27]. However, it is intuitively beneficial to account for the dissimilarity between nodes since it can signify complementary relationships among nodes, bearing significant implications for traffic condition modeling. To illustrate this, consider a traffic network where dissimilar nodes, such as a major highway and a narrow alley or a bustling urban intersection and a quiet suburban street, may exhibit contrasting traffic patterns. Capturing this dissimilarity can provide valuable insights into predicting traffic conditions accurately. Unfortunately, for each node, its dissimilar nodes are typically not participating in message passing, which could result in information loss and hence inferior performance. To tackle this issue, we provide both semantic affinity graphs and semantic penalty graphs to model diverse relationships in traffic networks.
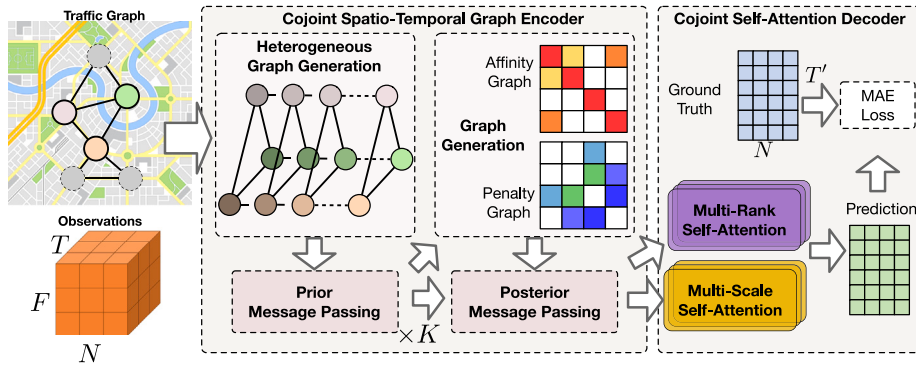
**Fig. 2.** Framework overview of the proposed COOL.

In detail, we first calculate the correlation scores between node pairs using cosine similarity. Formally, given the embeddings after neighboring aggregation on the road network $h_i^t$, we have:

$$s_{ij}^t = Score(h_i^t, h_j^t) = \phi\left(w \odot h_i^t, w \odot h_j^t\right), \tag{2}$$

where $w$ is a learnable vector to decide the importance of different dimensions and $\odot$ is the Hadamard product. $\phi(\cdot, \cdot)$ calculates the cosine similarity between two vectors. Note that the correlation scores could be positive or negative. On this basis, we construct a semantic affinity graph $W^t$ and a semantic penalty graph $P_{ij}^t$ at the $t$th step. Formally,

$$W_{ij}^t = \begin{cases} s_{ij}^t & s_{ij} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$P_{ij}^t = \begin{cases} -s_{ij}^t & s_{ij} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Similarly, we construct the heterogeneous version of two graphs containing observations between the time span $[t - r + 1, t]$ by aggregating the temporal information. In formulation, we have the following equations:

$$\hat{W}(v_i^t, v_j^{t'}) = \begin{cases} Score(h_i^t, h_j^{t'}) & Score(h_i^t, h_j^{t'}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$\hat{P}(v_i^t, v_j^{t'}) = \begin{cases} -Score(h_i^t, h_j^{t'}) & Score(h_i^t, h_j^{t'}) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Note that the affinity graphs and penalty graphs also play a crucial role in handling noise in graph construction to ensure robustness. (i) Affinity Graphs $W_{ij}^t$: These graphs capture positive correlations between node pairs based on cosine similarity scores. Nodes exhibiting similarity in traffic patterns are connected with non-zero weights, emphasizing their positive relationships. This helps in preserving meaningful connections in the graph. (ii) Penalty Graphs $P_{ij}^t$: These graphs focus on negative correlations between node pairs, nodes with dissimilar traffic patterns are connected with non-zero weights representing the dissimilarity. This is essential for explicitly considering contrasting relationships and preventing the loss of information due to dissimilar nodes. By incorporating both types of graphs in the model, the system addresses the limitations of traditional message passing that often neglects dissimilarity. This explicit modeling of positive and negative relationships enhances the robustness of the graph representation to noise, ensuring that the model can effectively capture the diverse dynamics of a traffic network. It enables a more accurate and resilient traffic condition prediction by considering both complementary and contrasting relationships among nodes.

**Posterior Message Passing.** Intuitively, the node representations connected in the affinity graph should be close while the ones connected in the penalty graph should be far away. To accomplish this,

we propose a correlation learning optimization objective as auxiliary loss function as follows:

$$\begin{aligned} \mathcal{L}_{cor} = &\sum_{t'=t-r+1}^{t} \sum_{i=1}^{N} \sum_{v_j^{t''} \in N_W(v_i^{t'})} \hat{W}(v_i^{t'}, v_j^{t''}) \gamma(u_i^{t'}, h_j^{t''}) \\ &- \sum_{t'=t-r+1}^{t} \sum_{i=1}^{N} \sum_{v_j^{t''} \in N_P(v_i^{t'})} \hat{P}(v_i^{t'}, v_j^{t''}) \gamma(u_i^{t'}, h_j^{t''}) \\ &+ \beta \sum_{t'=t-r+1}^{t} \sum_{i=1}^{N} \gamma(u_i^{t'}, h_i^{t'}), \end{aligned} \tag{7}$$

where $\{u_i^{t'}\}_{t' \in [t-r+1,t], i \in V}$ denotes the node representations to be optimized and $\gamma$ is a distance metric in the embedding space, $\beta$ is a hyperparameter used to balance the contributions of different losses (set to a default value of 1 in our experiments). The first term minimizes the distance between nodes connected in the affinity graph while the second term plays an opposite role for nodes connected in the penalty graph. The last term aims to reduce the variance of node representations for model stability. To facilitate optimization, $\gamma(\cdot, \cdot)$ is set to $l_2$-norm. Here, Eq. (7) has a *closed solution* by calculating the partial derivatives and the optimal node representations with normalization $u_i^{t'*}$ can be derived as follows:

$$u_i^{t'*} = \frac{h_i^{t'} + \sum_{v_j^{t''}} \hat{W}(v_i^{t'}, v_j^{t''}) h_j^{t''} - \sum_{v_j^{t''}} \hat{P}(v_i^{t'}, v_j^{t''}) h_j^{t''}}{\|h_i^{t'} + \sum_{v_j^{t''}} \hat{W}(v_i^{t'}, v_j^{t''}) h_j^{t''} - \sum_{v_j^{t''}} \hat{P}(v_i^{t'}, v_j^{t''}) h_j^{t''}\|_2}. \tag{8}$$

Through posterior message passing, we have incorporated similarities and dissimilarities into node representations for effective traffic forecasting. Finally, for each node, the optimal node representations are concatenated into a tensor $U_i = [u_i^1, \dots, u_i^T] \in \mathbb{R}^{T \times d}$ where $d$ is the embedding dimension.

### 3.2. Conjoint self-attention decoder

In reality, different locations in the traffic network could exhibit diverse transitional properties. For example, some locations could have periodic patterns like daily or monthly routines. To describe these diverse patterns in the traffic network, we offer a novel conjoint self-attention decoder, which sufficiently explores long-term correlations from both multi-rank and multi-scale perspectives.

**Multi-rank Self-Attention Branch.** Recently, Transformer has been extensively used to explore large-scale data in deep learning [55–60]. Inspired by this, we propose to utilize self-attention to identify long-term temporal relationships in traffic data. To decrease the parameters and avoid overfitting, we seek to utilize low-dimensional query vectors and value vectors for the low-rank attention matrix. Moreover, since various ranks may explore different features such as variances, our self-attention branch involves multiple ranks to generate multiple sequence embeddings.

**Algorithm 1** The overall learning algorithm of COOL.

**Input:** The traffic graph of road network $\mathcal{G}$, the traffic signals over the past time steps $X$;

**Output:** The prediction of traffic signals in future time steps;

1: Construct heterogeneous graph via Heterogeneous Graph Generator in Section 3.1;
2: **repeat**
3:     Compute each prior node representation $h_i^t$ by Eq. (1);
4:     Construct affinity and penalty graphs by Eqs. (3) and (4);
5:     Compute each posterior node representation $u_i^t$ by Eq. (8);
6:     Compute global embeddings $g$ by Eq. (11);
7:     Output the final prediction and calculate the MAE loss by Eq. (13);
8:     Backpropagation and update parameters by gradient descend;
9: **until** convergence

---

In detail, we take a given rank $\mu$ as an example. In this part, we omit the subscript of $U_i$ since spatial correlations are not considered. In this branch, for each node, the representation $u^t$ is converted into a query vector and a key vector, and their dot product is adopted to measure the importance of semantics at the current time step. To compress the embedding matrix, we introduce two low-dimensional left transformation matrices $\tilde{K}^\mu$ and $\tilde{V}^\mu \in \mathbb{R}^{(T/r) \times T}$, where $r$ is the number of the heads. Following the paradigm in Transformer [31], three extra right transformation matrices $\mathcal{Q}^\mu$, $\mathcal{K}^\mu$ and $\mathcal{V}^\mu \in \mathbb{R}^{d \times d}$ are defined to generate a query, a key and a value matrix. In formulation, we have:

$$\boldsymbol{\Gamma}^\mu = \mathrm{softmax}\left(\frac{\boldsymbol{U} \cdot \mathcal{Q}^\mu \left(\tilde{\boldsymbol{K}}^\mu \cdot \boldsymbol{U} \cdot \mathcal{K}^\mu\right)^\top}{\sqrt{d}}\right) \cdot \tilde{\boldsymbol{V}}^\mu \cdot \boldsymbol{U} \cdot \mathcal{V}^\mu, \qquad (9)$$

where $\boldsymbol{\Gamma}^\mu = [\boldsymbol{\gamma}^{\mu,1}, \ldots, \boldsymbol{\gamma}^{\mu,T}] \in \mathbb{R}^{T \times d}$. Finally, we summarize the embeddings for all time steps, producing rank-specific sequence embeddings $\boldsymbol{\gamma}^\mu \in \mathbb{R}^d$ and mean-pooling is adopted here. Similarly, for different ranks, we can obtain various rank-specific sequence embeddings. In our implementation, we choose three different ranks $\mu_1$, $\mu_2$ and $\mu_3$, which produce three sequence embeddings, i.e., $\boldsymbol{\gamma}^{\mu_1}$, $\boldsymbol{\gamma}^{\mu_2}$ and $\boldsymbol{\gamma}^{\mu_3}$ respectively.

**Multi-scale Self-attention Branch.** Furthermore, taking into account the potential periodic patterns in different locations, we propose a multi-scale self-attention branch. This branch involves pooling representation sequences at different scales and then leverages the self-attention mechanism to effectively fuse these subsequence embeddings.

Similarly, for each window size $\epsilon$ and each node, we can obtain subsequence embeddings by $\boldsymbol{\delta}^k = Pool(\boldsymbol{u}^{k\epsilon - \epsilon + 1}, \ldots, \boldsymbol{u}^{k\epsilon})$. Afterwards, the stacked matrix $\boldsymbol{\Delta}^\epsilon = [\boldsymbol{\delta}^1, \ldots, \boldsymbol{\delta}^{T/\epsilon}] \in \mathbb{R}^{T/\epsilon \times d}$ is fed into a self-attention module, which aggregates them into a scale-aware sequence embedding. Here, $\mathcal{Q}^\epsilon$, $\mathcal{K}^\epsilon$ and $\mathcal{V}^\epsilon \in \mathbb{R}^{d \times d}$ denotes transformation matrices of a query, a key and a value, respectively. Then, we calculate the weight using the dot-product to quantify the importance of each subsequence, and combine these subsequence representations to obtain a global representation. Formally, we have:

$$\boldsymbol{\Gamma}^\epsilon = \mathrm{softmax}\left(\frac{\boldsymbol{\Delta}^\epsilon \cdot \mathcal{Q}^\epsilon \left(\boldsymbol{\Delta}^\epsilon \cdot \mathcal{K}^\epsilon\right)^\top}{\sqrt{d}}\right) \cdot \boldsymbol{\Delta}^\epsilon \cdot \mathcal{V}^\epsilon. \qquad (10)$$

Similarly, with mean-pooling operation along the time dimension, we can generate the final global representation $\boldsymbol{\gamma}^\epsilon$. Again, three window sizes, i.e., $\epsilon_1$, $\epsilon_2$ and $\epsilon_3$ are selected, generating three embeddings from different views $\boldsymbol{\gamma}^{\epsilon_1}$, $\boldsymbol{\gamma}^{\epsilon_2}$ and $\boldsymbol{\gamma}^{\epsilon_3}$, respectively.

Finally, we introduce learnable parameters to aggregate these learned embeddings. In particular, with $\{w^{\mu_j}\}_{j=1}^3$ and $\{w^{\epsilon_j}\}_{j=1}^3$, and we can generate the final embedding $g$:

$$\boldsymbol{g} = \frac{\sum_{j=1}^3 \exp(w^{\mu_j})\boldsymbol{\gamma}^{\mu_j} + \exp(w^{\epsilon_j})\boldsymbol{\gamma}^{\epsilon_j}}{\sum_{j=1}^3 \exp(w^{\mu_j}) + \exp(w^{\epsilon_j})}. \qquad (11)$$

**Table 1**
Statistics of the evaluation datasets.

| Dataset | PEMS-BAY | PEMS08 | METR-LA | PEMS07 |
|---|---|---|---|---|
| $|\mathcal{V}|$ | 325 | 170 | 207 | 883 |
| $|\mathcal{E}|$ | 2369 | 295 | 1515 | 866 |
| Time steps | 52,116 | 17,856 | 34,272 | 28,224 |
| Time range | 01/01/2017–03/31/2017 | 07/01/2016–08/31/2016 | 03/01/2012–06/30/2012 | 05/01/2017–08/31/2017 |
| Interval | 5 min | 5 min | 5 min | 5 min |
| Place | San Francisco Bay Area, USA | San Bernardino Area, USA | Los Angeles, USA | California, USA |

The embedding $g$ would be combined with the state of the final step, i.e., $u^T$ to generate the predictions using a multi-layer perceptron (MLP) as:

$$\hat{Y} = \mathrm{MLP}\left(g \parallel u^T\right), \qquad (12)$$

where $\hat{Y}$ is the prediction, $\parallel$ denotes the concatenation operation. We optimize the overall framework by minimizing the standard mean absolute error (MAE) loss as follows:

$$\mathcal{L}_{mse} = \sum_t \left| Y_t - \hat{Y}_t \right|, \qquad (13)$$

The whole algorithm is summarized in Algorithm 1.

## 4. Experiments

### 4.1. Experimental setup

**Datasets and Metrics.** To comprehensively assess the performance of our proposed COOL, we conduct experiments on four real-world traffic datasets: *PEMS-BAY*, *PEMS08*, *METR-LA*, and *PEMS07*, and we summarize the statistics of datasets in Table 1.

For each of these datasets, we utilize sixty minutes of historical data to forecast traffic conditions for the subsequent sixty minutes. To evaluate the forecasting accuracy, we employ three common metrics: mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE), considering prediction horizons of 3, 6, and 12 time steps, providing a comprehensive assessment of COOL's predictive capabilities across different datasets and forecasting periods.

**Baseline Models.** To comprehensively evaluate the performance of our proposed COOL, we benchmark it against a diverse set of baseline models, encompassing both traditional methods and state-of-the-art neural network-based methods. Traditional methods involve:

- HA (Historical Average): HA is a baseline method for prediction that utilizes past data averages as a simple forecasting strategy.
- VAR (Vector Auto-Regressive) [61]: VAR is a time series model that extends autoregression to multiple variables, capturing interdependencies among them.
- SVR (Support Vector Regression) [62]: SVR employs a linear support vector machine for regression in the domain of classical time series analysis.

On the other hand, the neural network-based methods comprise:

- DCRNN [14]: DCRNN models traffic flow as a diffusion process on a directed graph, utilizing bidirectional random walks on the graph and an encoder–decoder architecture with scheduled sampling to capture spatiotemporal dependencies.
- STGCN [63]: STGCN employs a fully convolutional structure with a combination of graph convolutional layers and convolutional sequence learning layers to model spatial and temporal dependencies.

- ASTGCN [17]: ASTGCN adopts the spatio–temporal attention mechanism to capture the spatio–temporal correlation in traffic flow, and leverages graph convolution and standard convolution to mine static spatio–temporal features.
- STSGCN [18]: STSGCN designs spatial–temporal synchronous modeling mechanism and multiple modules for different time periods to capture the localized correlations and heterogeneities.
- MTGNN [26]: MTGNN proposes a novel mix-hop propagation layer and a dilated inception layer to capture the spatial and temporal dependencies, with the function of automatically extracting the uni-directed relations.
- GMAN [25]: GMAN incorporates multiple spatio–temporal attention blocks into the encoder–decoder architecture to model the impact of the spatio–temporal factors.
- DGCRN [27]: DGCRN leverages filtered node embeddings to generate a dynamic graph and combines it with the pre-defined static graph, which jointly facilitates topology modeling.
- DSTGCN [20]: DSTGCN performs similarity learning to extract the complex spatial–temporal relationships and utilizes a convolution module with an attention mechanism to dynamically extract spatial–temporal dependence.
- PDFormer [64]: PDFormer proposes a spatial self-attention module and a traffic delay-aware feature transformation module to respectively model dynamic spatial dependencies and the time delay of spatial information propagation.
- STAEformer [30]: STAEformer designs the spatio–temporal adaptive embedding to capture the intricate spatiotemporal traffic patterns and address the diminishing performance gains.

**Implemental Details.** Our proposed model COOL is optimized on an NVIDIA RTX GPU, capitalizing on its parallel processing capabilities to accelerate training and inference tasks. For our proposed model, we set the embedding dimension $d$ to 64 for the encoder. The encoder comprises six prior message-passing layers, enabling the model to process and propagate information across the graph structure efficiently. For the decoder, we carefully tune two critical components: ranks $[\mu_1, \mu_2, \mu_3]$ and window sizes $[\epsilon_1, \epsilon_2, \epsilon_3]$. Specifically, we choose to use values of $[3, 4, 6]$ for both ranks and window sizes. This configuration was found to be optimal and demonstrated robust performance across different datasets and prediction horizons. At the end of the decoder, a two-layer fully connected neural network is employed to transform the learned representations into the final prediction. To optimize the entire framework, we utilize the Adam optimizer [65], a widely used optimization algorithm that adapts learning rates during training. The model is trained for 100 epochs to ensure convergence, with a learning rate of 0.001 and a batch size of 32, all carefully chosen to balance training efficiency and model performance.

### 4.2. The performance of COOL

In our experiments, we follow the standard dataset division protocols commonly followed in previous works [27]. To ensure robust evaluation, we employ the following data split strategies for each of the datasets under consideration: For the PEMS-BAY and METR-LA datasets, we allocated 70% of the data for training, 10% for validation, and the remaining 20% for testing. For the PEMS08 dataset, we opt for a slightly different split, with 60% of the data used for training, 20% for validation, and the final 20% for testing. The results of our experiments compared with a range of different baselines, are presented in Table 2. We also provide more detailed results on the PEMS07 dataset in Table 3, where we present the results of 12 horizons. From the results, we have three observations as follows:

- Generally, traditional methods showcase inferior performance when compared to cutting-edge neural network-based approaches. This performance gap can be attributed to the fact that traditional methods predominantly consider temporal correlations while neglecting critical spatial dependencies within the data.

- Our proposed COOL achieves the best performance across a majority of settings and datasets, underscoring the effectiveness of our novel framework. Note that PDFormer [64] and STAEformer [30], both based on the transformer mechanism, exhibit slightly superior performance on the PEMS08 dataset for Horizon = 3 and 6 predictions. This may be attributed to the ability of spatiotemporal transformers to effectively capture dependencies over medium to long time intervals. However, as the time horizon increases (Horizon = 12), the superiority of our approach in long-range predictions becomes more pronounced. This notable success can be primarily attributed to two key factors. First, our proposed COOL leverages a conjoint spatio–temporal graph encoder, which facilitates the exploration of higher-order relations, making it adept at capturing complex dependencies within the traffic flow data. Second, our model incorporates multi-rank and multi-scale self-attention branches, enabling it to effectively capture a diverse range of sequential trends.
- To demonstrate the superiority of our proposed COOL across diverse horizons, we compare it with state-of-the-art baselines (i.e., PDFormer and STAEformer) on the PEMS07 dataset. We evaluate the performance using MAPEs as metrics for different horizons ranging from Horizon 1 to 12. The results, shown in Table 3, reveal that our model consistently outperforms the competitive methods at all different horizons compared to the baselines. This clearly highlights the outstanding capability of our proposed model in exploring spatio–temporal dependencies.
- Remarkably, our model exhibit the most substantial improvement on the PEMS-BAY and PEMS07 datasets, which are among the largest and most challenging datasets considered. This underscores the scalability and robustness of our model when faced with large-scale traffic prediction tasks.

### 4.3. Ablation studies

In this subsection, we thoroughly conduct ablation studies to assess the efficacy of each constituent component incorporated within our proposed COOL framework. Our evaluation is performed on both the PEMS08 and METR-LA datasets. To conduct these ablations, we systematically remove individual components, including prior graph convolution, posterior graph convolution, multi-rank self-attention, and multi-scale self-attention, from our COOL model. Subsequently, we evaluate the performance of the modified models with these components removed.

The results of these ablation experiments are carefully reported in Table 4. As can be seen from the results, the removal of any single component inevitably leads to a noticeable performance degradation. This compelling evidence highlights the pivotal effectiveness of each module in our framework. It is noteworthy that removing either the prior graph convolution or the posterior graph convolution in isolation does not significantly impact the results. This suggests a degree of complementarity between these two graph convolution components, further illustrating the inherent robustness of the COOL model.

### 4.4. Hyperparameter analysis

In this part, we investigate the sensitivity of the model's hyperparameters, with a specific focus on the different ranks and window sizes employed in multi-rank and multi-scale self-attention mechanisms, respectively. Our findings are thoroughly presented in Fig. 3, where we distinguish between experiments conducted on the PEMS08 dataset (depicted in Figs. (a) and (b)) and those carried out on METR-LA (illustrated in Figs. (c) and (d)). Figs. (a) and (c) delve into the influence of hyperparameters $\mu_1$, $\mu_2$, and $\mu_3$ in the context of multi-rank self-attention, while Figs. (b) and (d) analyze the impact of $\epsilon_1$, $\epsilon_2$, and $\epsilon_3$ within the realm of multi-scale self-attention. Generally, our proposed

**Table 2**
Traffic forecasting results on the PEMS-BAY, PEMS08 and METR-LA datasets.

| Datasets | Methods | Horizon 3 | | | Horizon 6 | | | Horizon 12 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| PEMS-BAY | HA | 1.89 | 4.30 | 4.16% | 2.50 | 5.82 | 5.62% | 3.31 | 7.54 | 7.65% |
| | VAR | 1.74 | 3.16 | 3.60% | 2.32 | 4.25 | 5.00% | 2.93 | 5.44 | 6.50% |
| | SVR | 1.85 | 3.59 | 3.80% | 2.48 | 5.18 | 5.50% | 3.28 | 7.08 | 8.00% |
| | DCRNN | 1.38 | 2.95 | 2.90% | 1.74 | 3.97 | 3.90% | 2.07 | 4.74 | 4.90% |
| | STGCN | 1.36 | 2.96 | 2.90% | 1.81 | 4.27 | 4.17% | 2.49 | 5.69 | 5.79% |
| | ASTGCN | 1.52 | 3.13 | 3.22% | 2.01 | 4.27 | 4.48% | 2.61 | 5.42 | 6.00% |
| | STSGCN | 1.44 | 3.01 | 3.04% | 1.83 | 4.18 | 4.17% | 2.26 | 5.21 | 5.40% |
| | MTGNN | 1.32 | 2.79 | 2.77% | 1.65 | 3.74 | 3.69% | 1.94 | 4.49 | 4.53% |
| | GMAN | 1.34 | 2.91 | 2.86% | 1.63 | 3.76 | 3.68% | 1.86 | 4.32 | 4.37% |
| | DGCRN | 1.28 | 2.69 | 2.66% | 1.59 | 3.63 | 3.55% | 1.89 | 4.42 | 4.43% |
| | DSTGCN | 1.17 | 2.30 | 2.37% | 1.60 | 3.54 | 3.63% | 1.94 | 4.39 | 4.65% |
| | PDFormer | 1.16 | 2.31 | 2.36% | 1.61 | 3.61 | 3.58% | 1.96 | 4.45 | 4.55% |
| | STAEformer | 1.30 | 2.78 | 2.76% | 1.61 | 3.69 | 3.62% | 1.88 | 4.34 | 4.37% |
| | COOL | **1.13** | **2.23** | **2.29%** | **1.53** | **3.42** | **3.38%** | **1.84** | **4.23** | **4.28%** |
| PEMS08 | HA | 23.52 | 34.96 | 14.72% | 27.67 | 40.89 | 17.37% | 39.28 | 56.74 | 25.17% |
| | VAR | 19.52 | 29.73 | 12.54% | 22.25 | 33.30 | 14.23% | 26.17 | 38.97 | 17.32% |
| | SVR | 17.93 | 27.69 | 10.95% | 22.41 | 34.53 | 13.97% | 32.11 | 47.03 | 20.99% |
| | DCRNN | 15.64 | 25.48 | 10.04% | 17.88 | 27.63 | 11.38% | 22.51 | 34.21 | 14.17% |
| | STGCN | 15.30 | 25.03 | 9.88% | 17.69 | 27.27 | 11.03% | 25.46 | 33.71 | 13.34% |
| | ASTGCN | 16.48 | 25.09 | 11.03% | 18.66 | 28.17 | 12.23% | 22.83 | 33.68 | 15.24% |
| | STSGCN | 15.45 | 24.39 | 10.22% | 16.93 | 26.53 | 10.84% | 19.50 | 30.43 | 12.27% |
| | MTGNN | 14.24 | 22.43 | 9.02% | 15.30 | 24.32 | 9.58% | 16.85 | 26.93 | 10.57% |
| | GMAN | 13.80 | 22.88 | 9.41% | 14.62 | 24.02 | 9.57% | 15.72 | 25.96 | 10.56% |
| | DGCRN | 13.89 | 22.07 | 9.19% | 14.92 | 23.99 | 9.85% | 16.73 | 26.88 | 10.84% |
| | DSTGCN | 13.65 | 21.86 | 10.26% | 14.50 | 23.67 | 11.48% | 15.78 | 26.09 | 12.50% |
| | PDFormer | 13.02 | 22.25 | 8.70% | 13.81 | 24.02 | 9.14% | 15.59 | 26.09 | 10.41% |
| | STAEformer | **12.91** | 22.17 | **8.61%** | **13.67** | 23.71 | **9.00%** | 15.17 | 25.87 | 10.08% |
| | COOL | 13.18 | **21.87** | 8.82% | 13.98 | **23.65** | 9.18% | **15.07** | **25.93** | **9.81%** |
| METR-LA | HA | 4.79 | 10.00 | 11.70% | 5.47 | 11.45 | 13.50% | 6.99 | 13.89 | 17.54% |
| | VAR | 4.42 | 7.80 | 13.00% | 5.41 | 9.13 | 12.70% | 6.52 | 10.11 | 15.80% |
| | SVR | 3.39 | 8.45 | 9.30% | 5.05 | 10.87 | 12.10% | 6.72 | 13.76 | 16.70% |
| | DCRNN | 2.77 | 5.38 | 7.30% | 3.15 | 6.45 | 8.80% | 3.60 | 7.60 | 10.50% |
| | STGCN | 2.88 | 5.74 | 7.62% | 3.47 | 7.24 | 9.57% | 4.59 | 9.40 | 12.70% |
| | ASTGCN | 4.86 | 9.27 | 9.21% | 5.43 | 10.61 | 10.13% | 6.51 | 12.52 | 11.64% |
| | STSGCN | 3.31 | 7.62 | 8.06% | 4.13 | 9.77 | 10.29% | 5.06 | 11.66 | 12.91% |
| | MTGNN | 2.69 | 5.18 | 6.88% | 3.05 | 6.17 | 8.19% | 3.49 | 7.23 | 9.87% |
| | GMAN | 2.80 | 5.55 | 7.41% | 3.12 | 6.49 | 8.73% | 3.44 | 7.35 | 10.07% |
| | DGCRN | 2.62 | 5.01 | 6.63% | **2.99** | 6.05 | **8.02%** | 3.44 | 7.19 | 9.73% |
| | DSTGCN | 2.68 | 4.97 | 7.21% | 3.12 | 6.18 | 9.02% | 3.50 | 7.19 | 10.65% |
| | PDFormer | 2.83 | 5.59 | 7.26% | 4.05 | 6.59 | 10.32% | 4.80 | 7.82 | 11.20% |
| | STAEformer | 2.65 | 5.11 | 6.86% | 3.04 | 6.01 | 8.15% | 3.50 | 7.12 | 9.83% |
| | COOL | **2.57** | **4.70** | **6.43%** | 3.01 | **5.96** | 8.13% | **3.43** | **7.07** | **9.70%** |

**Table 3**
Comparison of methods across different horizons (from H 1 to H 12). The experiments are performed on PEMS07 dataset and MAPEs are shown.

| Method | H 1 | H 2 | H 3 | H 4 | H 5 | H 6 | H 7 | H 8 | H 9 | H 10 | H 11 | H 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COOL | 1.16% | 7.27% | 7.78% | 7.72% | 7.85% | 8.03% | 8.14% | 8.31% | 8.47% | 8.60% | 8.66% | 8.76% |
| STAEformer | 6.96% | 7.30% | 7.55% | 7.75% | 7.91% | 8.06% | 8.20% | 8.36% | 8.48% | 8.63% | 8.76% | 8.97% |
| PDFormer | 7.99% | 8.25% | 8.53% | 8.70% | 8.94% | 9.09% | 9.14% | 9.37% | 9.46% | 9.66% | 9.84% | 10.42% |

**Table 4**
Ablated study on PEMS08 and METR-LA datasets. The performance is measured in terms of MAE.

| PEMS08 | Horizon 3 | Horizon 6 | Horizon 12 |
|---|---|---|---|
| COOL | **13.18** | **13.98** | **15.07** |
| W/o Prior | 13.21 | 14.04 | 15.26 |
| W/o Posterior | 13.26 | 14.09 | 15.19 |
| W/o Multi-rank | 13.37 | 14.28 | 15.43 |
| W/o Multi-scale | 13.34 | 14.10 | 15.25 |

| METR-LA | Horizon 3 | Horizon 6 | Horizon 12 |
|---|---|---|---|
| COOL | **2.57** | **3.01** | **3.43** |
| W/o Prior | 2.64 | 3.10 | 3.51 |
| W/o Posterior | 2.60 | 3.05 | 3.46 |
| W/o Multi-rank | 2.59 | 3.04 | 3.45 |
| W/o Multi-scale | 2.59 | 3.03 | 3.44 |

COOL exhibits a robustness to variations in these hyperparameters. However, it is noteworthy that a combination of window sizes $[3, 4, 6]$ consistently yields slightly superior performance in both the multi-rank and multi-scale self-attention branches. This phenomenon might be attributed to the fact that medium-sized window sizes exhibit an enhanced capability to capture the subtle variations in traffic conditions, as opposed to larger window sizes.

### 4.5. Efficiency analysis

To evaluate our model's performance efficiency against state-of-the-art baseline models. Specifically, we compare our model against PDFormer and STAEformer in terms of the number of parameters, the duration of training per epoch, and the testing time. The results shown in Table 5 clearly demonstrate that our model COOL operates with a significantly lower number of parameters, indicating a higher efficiency in terms of model size compared to the baselines. Besides,
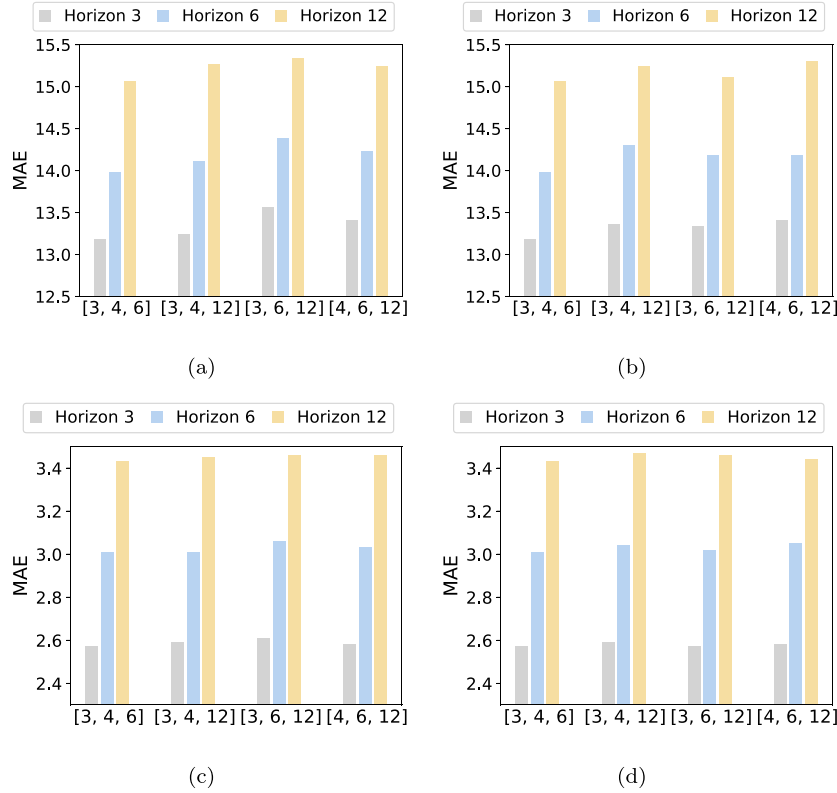
**Fig. 3.** Hyperparameter study of the proposed COOL on PEMS08 and METR-LA.

**Table 5**
Comparison of model efficiency.

| Model | Number of parameters | Training time per epoch | Test time |
|---|---|---|---|
| COOL | 264K | 87.3 s | 5.4 s |
| PDFormer | 531K | 86.9 s | 9.3 s |
| STAEFormer | 1123K | 91.0 s | 6.6 s |

COOL's training time is comparable to that of the baseline models, showcasing its efficiency in learning. Moreover, our COOL achieves a shorter test time than both baselines, underscoring its superior performance in terms of inference speed and operational efficiency. This highlights COOL's effectiveness in balancing model complexity with computational efficiency, making it a highly competitive choice for applications requiring rapid predictions.

### 4.6. Visualization of prediction results

In this part, we provide the visualization of forecasting results. Fig. 4 visualizes the prediction results and ground truth values of sensors No. 19 and No. 194 from Jun 13 to Jun 15. Except for some noises (maybe caused by sudden accidents or sensor failures), the proposed COOL predicts the traffic condition with impressive accuracy. In sensor No. 19, the pattern is regular: every evening the traffic speed drops due to congestion. The model easily captures this pattern and provides a good prediction. In sensor No. 194, the traffic condition is more diverse and complicated: the traffic speed drops significantly in the afternoon, but the starting time and duration of this congestion are very different over the three days. On the first day, this afternoon congestion appears late and is quickly over, whereas on the last day it comes early and lasts longer. By handling the diverse traffic conditions, our proposed COOL successfully predicts the traffic speed with reasonable accuracy under this complicated situation, validating the superiority of our method.
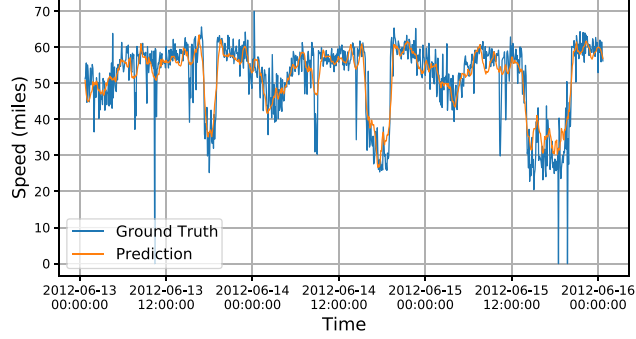
### 4.7. Visualization of learned attentions

In this subsection, we provide the visualization of learned attention. More specifically, we visualize the attention scores multi-scale self-attention of the proposed COOL. The experiments are performed on METR-LA dataset and the learned attentions of multiple scales are visualized using the heatmaps. The experimental results are shown in Fig. 5. As can be seen from the results, the attention of different scales exhibits different patterns which shows that the proposed attention module can capture meaningful information at different scales. For example, in Fig. 5(a), we use a window size of 4 and the attention distributed uniformly within the inputs. In comparison, when the window size is set to 3 (which is shown in Fig. 5(b)), the first window captures meaningful information that is attended by other inputs. This shows that arranging the temporal inputs at different scales and learn the attended output thereafter is a way to view the data from different perspectives and are more likely to capture essential information at specific scales.

### 4.8. Case study

This section provides a case study to showcase the model's ability to capture spatio–temporal dependencies and diverse transitions, and the results are shown in Fig. 6. The left of the figure shows the affinity graph matrix, and the right of the figure shows the traffic flow data of two sensors (*i.e.,* sensor 9 and sensor 10). As can be seen from the visualization of the affinity matrix, sensor 9 and sensor 10 (indicated by the 10th row and 9th column of the matrix) have very high affinity scores, which shows that they are highly related in terms of traffic flow features. The right part of the figure demonstrates this, in which the traffic flow of sensor 9 and sensor 10 shows similar patterns and the traffic flow data of sensor 9 is one step behind the data of sensor 10. This shows that our model can capture spatial and temporal correlations in the traffic flow data.
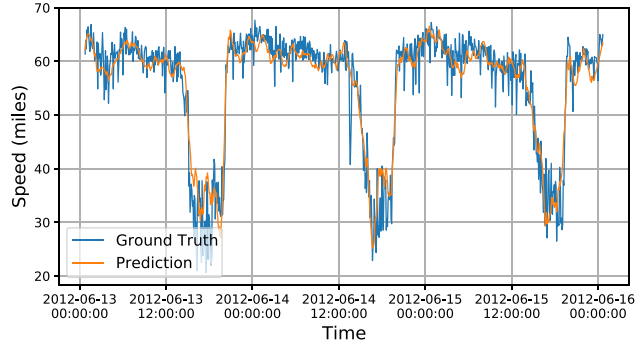
(a) Visualization on sensor No.19



(b) Visualization on sensor No.194

**Fig. 4.** Visualization of prediction results on METR-LA.



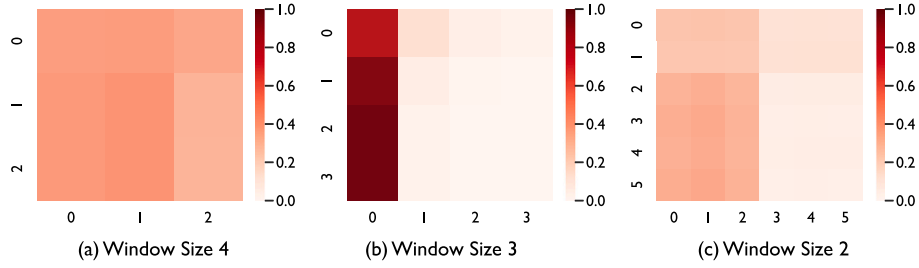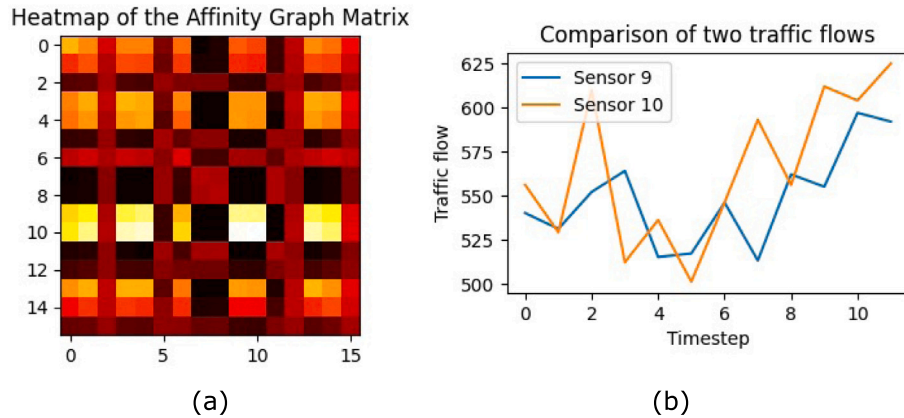(a) Window Size 4      (b) Window Size 3      (c) Window Size 2

**Fig. 5.** Visualization of learned attentions scores of multi-scale attention module. Sub-figure (a) is the attention scores of window size 4, Sub-figure (b) is the attention scores of window size 3, Sub-figure (c) is the attention scores of window size 2.



(a)            (b)

**Fig. 6.** The case study of the proposed model. The left (a) is the affinity graph matrix, while the right (b) is the traffic flow data of two sensors (*i.e.,* sensor 9 and sensor 10).

## 4.9. Potential limitations and drawbacks

**Limited Exploration of Prior Information.** While the COOL introduces prior information into heterogeneous graphs, the effectiveness of capturing high-order spatio–temporal relationships from this information may be limited. The model's performance might be sensitive to the quality and relevance of the introduced prior knowledge, and the impact of different types of prior information on forecasting accuracy needs to be further explored.

**Dependency on Constructed Affinity and Penalty Graphs.** The reliance on constructed affinity and penalty graphs to model dynamic relationships introduces an additional layer of complexity. The performance of the COOL may be influenced by the accuracy of constructing these graphs, and the approach's robustness to variations in data quality or noise in the construction process should be investigated.

**Adaptability to New Traffic Scenarios.** The effectiveness of the proposed COOL in capturing diverse transitional patterns is demonstrated on benchmark datasets. However, its adaptability to new or unseen traffic scenarios, such as emerging traffic patterns in rapidly changing urban environments, is an open question and requires further investigation.

## 5. Conclusion

This paper studies traffic flow forecasting and proposes a novel method named COOL to solve it. Our proposed COOL conjointly explores high-order spatio–temporal relationships from both prior and posterior information. In particular, we not only extend road networks into heterogeneous graphs for prior message passing, but also model dynamic relationships using both affinity graphs and penalty graphs for posterior message passing. Moreover, we develop a conjoint self-attention decoder to capture diverse temporal properties in traffic data. Experimental results on three traffic datasets demonstrate the superiority of our proposed model COOL, which outperforms the state-of-the-art baselines.

In the future, we plan to extend our GNN-based traffic forecasting model by incorporating additional data sources, such as real-time weather and events data, to further improve prediction accuracy. Additionally, we aim to explore the integration of reinforcement learning techniques to optimize traffic signal control and reduce congestion. Furthermore, investigating the scalability of our model to larger urban networks and evaluating its robustness under various traffic conditions are important directions for future research. Finally, we will continue to explore novel methods for interpretability and visualization of the model's predictions to enhance its practical utility for urban traffic management.

## CRediT authorship contribution statement

**Wei Ju:** Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Yusheng Zhao:** Writing – original draft, Software, Methodology, Conceptualization. **Yifang Qin:** Software. **Siyu Yi:** Software, Writing – review & editing. **Jingyang Yuan:** Software. **Zhiping Xiao:** Writing – review & editing, Software. **Xiao Luo:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Xiting Yan:** Writing – review & editing, Supervision, Project administration. **Ming Zhang:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## References

[1] X. Wu, D. Zhang, C. Guo, C. He, B. Yang, C.S. Jensen, AutoCTS: Automated correlated time series forecasting, Proc. VLDB Endow. 15 (4) (2021) 971–983.

[2] H. Li, Y. Zhao, Z. Mao, Y. Qin, Z. Xiao, J. Feng, Y. Gu, W. Ju, X. Luo, M. Zhang, A survey on graph neural networks in intelligent transportation systems, 2024, arXiv preprint arXiv:2401.00713.

[3] J. Ji, J. Wang, Z. Jiang, J. Jiang, H. Zhang, STDEN: Towards physics-guided neural networks for traffic flow prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4048–4056.

[4] P. Xie, T. Li, J. Liu, S. Du, X. Yang, J. Zhang, Urban flow prediction from spatiotemporal data using machine learning: A survey, Inf. Fusion 59 (2020) 1–12.

[5] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, W. Lv, Generic dynamic graph convolutional network for traffic flow forecasting, Inf. Fusion 100 (2023) 101946.

[6] Y. Zhao, X. Luo, W. Ju, C. Chen, X.-S. Hua, M. Zhang, Dynamic hypergraph structure learning for traffic flow forecasting, in: 2023 IEEE 39th International Conference on Data Engineering, ICDE, IEEE, 2023, pp. 2303–2316.

[7] A.A. Ouallane, A. Bakali, A. Bahnasse, S. Broumi, M. Talea, Fusion of engineering insights and emerging trends: Intelligent urban traffic management system, Inf. Fusion 88 (2022) 218–248.

[8] Z. Luo, L. Li, M. Zhang, W. Hua, Y. Xu, X. Zhou, Diversified top-k route planning in road network, Proc. VLDB Endow. 15 (11) (2022) 3199–3212.

[9] D. Ni, Traffic Flow Theory: Characteristics, Experimental Methods, and Numerical Techniques, Butterworth-Heinemann, 2015.

[10] X. Di, R. Shi, A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning, Transp. Res. C 125 (2021) 103008.

[11] Z. Mo, R. Shi, X. Di, A physics-informed deep learning paradigm for car-following models, Transp. Res. C 130 (2021) 103240.

[12] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.

[13] Z. Sun, G. Fox, Traffic flow forecasting based on combination of multidimensional scaling and SVM, Int. J. Intell. Transp. Syst. Res. 12 (2014) 20–25.

[14] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2017, arXiv preprint arXiv:1707.01926.

[15] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, 2019, arXiv preprint arXiv:1906.00121.

[16] Q. Ren, Y. Li, Y. Liu, Transformer-enhanced periodic temporal convolution network for long short-term traffic flow forecasting, Expert Syst. Appl. 227 (2023) 120203.

[17] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 922–929.

[18] C. Song, Y. Lin, S. Guo, 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, 2020, pp. 914–921.

[19] Z. Fang, Q. Long, G. Song, K. Xie, Spatial-temporal graph ode networks for traffic flow forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2021, pp. 364–373.

[20] M. Liu, G. Liu, L. Sun, Spatial–temporal dependence and similarity aware traffic flow forecasting, Inform. Sci. 625 (2023) 81–96.

[21] K. Pholsena, L. Pan, Z. Zheng, Mode decomposition based deep learning model for multi-section traffic prediction, World Wide Web 23 (2020) 2513–2527.

[22] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, B. Yin, Metro passenger flow prediction via dynamic hypergraph convolution networks, IEEE Trans. Intell. Transp. Syst. 22 (12) (2021) 7891–7903.

[23] S. Yang, B. Yang, An inductive heterogeneous graph attention-based multi-agent deep graph infomax algorithm for adaptive traffic signal control, Inf. Fusion 88 (2022) 249–262.

[24] F.T. Sáenz, F. Arcas-Tunez, A. Muñoz, Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data, Inf. Fusion 91 (2023) 582–597.

[25] C. Zheng, X. Fan, C. Wang, J. Qi, GMAN: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 1234–1241.

[26] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2020, pp. 753–763.

[27] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, Y. Li, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, ACM Trans. Knowl. Discov. Data 17 (1) (2023) 1–21.

[28] L. Qu, W. Li, W. Li, D. Ma, Y. Wang, Daily long-term traffic flow forecasting based on a deep neural network, Expert Syst. Appl. 121 (2019) 304–312.

[29] X. Luo, D. Li, Y. Yang, S. Zhang, et al., Spatiotemporal traffic flow prediction with KNN and LSTM, J. Adv. Transp. 2019 (2019).

[30] H. Liu, Z. Dong, R. Jiang, J. Deng, J. Deng, Q. Chen, 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.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[32] W. Ju, Z. Fang, Y. Gu, Z. Liu, Q. Long, Z. Qiao, Y. Qin, J. Shen, F. Sun, Z. Xiao, et al., A comprehensive survey on deep graph representation learning, 2023, arXiv preprint arXiv:2304.05055.

[33] A. Holzinger, B. Malle, A. Saranti, B. Pfeifer, Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI, Inf. Fusion 71 (2021) 28–37.

[34] F. Zhou, T. Wang, T. Zhong, G. Trajcevski, Identifying user geolocation with hierarchical graph neural networks and explainable fusion, Inf. Fusion 81 (2022) 1–13.

[35] X. Luo, J. Yuan, Z. Huang, H. Jiang, Y. Qin, W. Ju, M. Zhang, Y. Sun, HOPE: High-order graph ODE for modeling interacting dynamics, in: International Conference on Machine Learning, PMLR, 2023, pp. 23124–23139.

[36] W.-C. Huang, C.-T. Chen, C. Lee, F.-H. Kuo, S.-H. Huang, Attentive gated graph sequence neural network-based time-series information fusion for financial trading, Inf. Fusion 91 (2023) 261–276.

[37] L.A. Passos, J.P. Papa, J. Del Ser, A. Hussain, A. Adeel, Multimodal audio-visual information fusion using canonical-correlated graph neural network for energy-efficient speech enhancement, Inf. Fusion 90 (2023) 1–11.

[38] W. Ju, S. Yi, Y. Wang, Q. Long, J. Luo, Z. Xiao, M. Zhang, A survey of data-efficient graph learning, 2024, arXiv preprint arXiv:2402.00447.

[39] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.

[40] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[41] Z. Fang, L. Xu, G. Song, Q. Long, Y. Zhang, Polarized graph neural networks, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1404–1413.

[42] X. Luo, W. Ju, Y. Gu, Y. Qin, S. Yi, D. Wu, L. Liu, M. Zhang, Toward effective semi-supervised node classification with hybrid curriculum pseudo-labeling, ACM Trans. Multimed. Comput. Commun. Appl. 20 (3) (2023) 1–19.

[43] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, J. Leskovec, Hierarchical graph representation learning with differentiable pooling, in: Advances in Neural Information Processing Systems, 2018, pp. 4805–4815.

[44] J. Lee, I. Lee, J. Kang, Self-attention graph pooling, in: International Conference on Machine Learning, PMLR, 2019, pp. 3734–3743.

[45] X. Luo, Y. Zhao, Z. Mao, Y. Qin, W. Ju, M. Zhang, Y. Sun, RIGNN: A rationale perspective for semi-supervised open-world graph classification, Trans. Mach. Learn. Res. (2023).

[46] X. Luo, Y. Zhao, Y. Qin, W. Ju, M. Zhang, Towards semi-supervised universal graph classification, IEEE Trans. Knowl. Data Eng. 36 (1) (2023) 416–428.

[47] J. Yuan, X. Luo, Y. Qin, Y. Zhao, W. Ju, M. Zhang, Learning on graphs under label noise, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[48] W. Ju, X. Luo, M. Qu, Y. Wang, C. Chen, M. Deng, X.-S. Hua, M. Zhang, TGNN: A joint semi-supervised framework for graph-level classification, 2023, arXiv preprint arXiv:2304.11688.

[49] L. Yue, X. Jun, Z. Sihang, W. Siwei, G. Xifeng, Y. Xihong, L. Ke, T. Wenxuan, L.X. Wang, et al., A survey of deep graph clustering: Taxonomy, challenge, and application, 2022, arXiv preprint arXiv:2211.12875.

[50] W. Ju, Y. Gu, B. Chen, G. Sun, Y. Qin, X. Liu, X. Luo, M. Zhang, GLCC: A general framework for graph-level clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 4391–4399.

[51] S. Yi, W. Ju, Y. Qin, X. Luo, L. Liu, Y. Zhou, M. Zhang, Redundancy-free self-supervised relational learning for graph clustering, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[52] C. Meng, S. Rambhatla, Y. Liu, Cross-node federated graph neural network for spatio–temporal data modeling, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2021, pp. 1202–1211.

[53] Z. Shao, Z. Zhang, F. Wang, Y. Xu, Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1567–1577.

[54] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, J. Zhang, Urban traffic prediction from spatio–temporal data using deep meta learning, in: Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2019, pp. 1720–1730.

[55] L. Xia, C. Huang, C. Zhang, Self-supervised hypergraph transformer for recommender systems, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2100–2109.

[56] Y. Yang, C. Huang, L. Xia, Y. Liang, Y. Yu, C. Li, Multi-behavior hypergraph-enhanced transformer for sequential recommendation, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 2263–2274.

[57] J. Zou, E. Kanoulas, P. Ren, Z. Ren, A. Sun, C. Long, Improving conversational recommender systems via transformer-based sequential modelling, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2319–2324.

[58] D. Liu, G.P. Farajalla, A. Boulenger, Transformer-based banking products recommender system, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 2641–2642.

[59] D. He, Y. Zhao, J. Luo, T. Hui, S. Huang, A. Zhang, S. Liu, Transrefer3D: Entity-and-relation aware transformer for fine-grained 3D visual grounding, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 2344–2352.

[60] Y. Zhao, J. Chen, C. Gao, W. Wang, L. Yang, H. Ren, H. Xia, S. Liu, Target-driven structured transformer planner for vision-language navigation, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 4194–4203.

[61] H. Lütkepohl, New Introduction to Multiple Time Series Analysis, Springer Science & Business Media, 2005.

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

[63] B. Yu, H. Yin, Z. Zhu, Spatio–temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2017, arXiv preprint arXiv:1709.04875.

[64] J. Jiang, C. Han, W.X. Zhao, 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, 2023, pp. 4365–4373.

[65] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.