Enhancing Spatiotemporal Traffic Prediction through Urban Human Activity Analysis

Sumin Han
School of Computing, KAIST
Republic of Korea
hsm6911@kaist.ac.kr

Youngjun Park
School of Computing, KAIST
Republic of Korea
youngjourpark@kaist.ac.kr

Minji Lee
School of Computing, KAIST
Republic of Korea
haewon_lee@kaist.ac.kr

Jisun An
Indiana University Bloomington (IUB)
United States
jisunan@iu.edu

Dongman Lee
School of Computing, KAIST
Republic of Korea
dlee@cs.kaist.ac.kr

ABSTRACT

Traffic prediction is one of key elements to ensure the safety and convenience of citizens. Existing traffic prediction models primarily focus on deep learning architectures to capture spatial and temporal correlation. They often overlook the underlying nature of traffic. Specifically, the sensor networks in most traffic datasets do not accurately represent the actual road network exploited by vehicles, failing to provide insights into the traffic patterns in urban activities. To overcome these limitations, we propose an improved traffic prediction method based on graph convolution deep learning algorithms. We leverage human activity frequency data from National Household Travel Survey to enhance the inference capability of a causal relationship between activity and traffic patterns. Despite making minimal modifications to the conventional graph convolutional recurrent networks and graph convolutional transformer architectures, our approach achieves the state-of-the-art performance without introducing excessive computational overhead.

CCS CONCEPTS

• **Computer systems organization** → **Neural networks**.

KEYWORDS

Spatiotemporal traffic prediction, Urban human activity, Graph convolutional network, Recurrent neural networks, Transformer

ACM Reference Format:
Sumin Han, Youngjun Park, Minji Lee, Jisun An, and Dongman Lee. 2018. Enhancing Spatiotemporal Traffic Prediction through Urban Human Activity Analysis. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Traffic prediction plays a crucial role in ensuring the safety and convenience of citizens by accurately estimating future traffic speed or volume based on historical patterns from sensor data. Unlike typical time-series prediction problems, traffic prediction requires inferring a sensor's traffic values by leveraging patterns observed in other sensors. Previous studies have explored various spatiotemporal models based on Recurrent Neural Networks (RNN) [11, 18, 31, 32] and Transformer [12, 15, 25], which have shown effectiveness in time series prediction while incorporating the spatial similarity between traffic sensors.

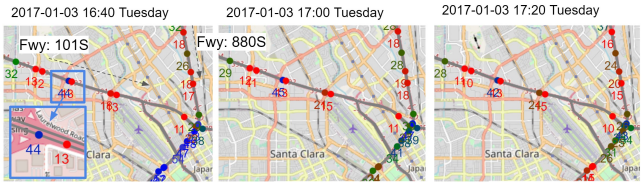


Figure 1: Congestion pattern of sensors on specific freeways in PEMS-BAY dataset (congestion level: red>green>blue).

We argue that there still remain key points for improving traffic prediction:

1. Construction of an accurate graph representation of the sensor network: Existing studies [18, 31, 32] construct adjacency matrices based on sensor proximity using distance thresholds. Although some models [12, 24, 25, 30] have attempted to optimize the sensor adjacency matrix by learning from data, these approaches often result in inefficiency and high computational costs. Additionally, they may generate adjacency matrices that include artificial connections, leading to inaccuracies and incorrect representations of the sensor network. For example, two closely located traffic sensors may assess different traffic patterns due to their positioning on different road directions.

2. Addressing individual sensor spatial heterogeneity: Each traffic sensor is situated within a unique built environment, resulting in diverse congestion patterns. For instance, congestion due to rush hour may occur only in specific lanes or sensors. Even in close locations, different patterns can emerge due to factors such as the number of sensor lanes, entry and exit lanes, and installation positions. Fig. 1 illustrates a significant discrepancy in values between adjacent sensors, where a sensor recording a speed of 44 miles per hour (mph) can be considered congested relative to neighboring sensors. While previous work such as [12] has addressed this issue by leveraging spatial positional encoding for Transformer models, the primary focus has been on handling positional encoding rather than normalizing the patterns of individual sensors.

CIKM '23, October 21–25, 2023, Bermingham, United Kingdom

3. Incorporating human activity-based inference for traffic prediction: Human activities, such as commuting, significantly influence traffic patterns and can lead to congestion. Previous studies have incorporated temporal information such as weekday and time-of-day to capture correlations between time and traffic patterns [12, 15, 33]. While temporal information provides insights into human activities, it does not establish direct causality, as traffic patterns are influenced by human actions. Indirectly inferring human actions from temporal information makes it challenging for models to detect variations in behavior, such as during holidays or seasonal activity difference, or learn about similar behaviors occurring at different times.

In this study, we propose a novel solution to address the challenges of generating realistic vehicle travel trajectories while ensuring sensor connectivity. We utilize the A* algorithm to generate these trajectories and derive pairwise similarity measures between sensors, integrating them into graph convolutional temporal models. To accommodate the spatial heterogeneity of sensors, we employ a one-hot-based sensor encoding specific to each sensor, enabling adaptability to diverse sensor environments. To capture the correlation between human activity and traffic patterns, we incorporate urban travel activity frequencies from the National Household Travel Survey[28] estimated at the target prediction timestamp. Our approach consists of two spatiotemporal deep learning architectures, namely UAGCRN and UAGCTransformer, which effectively integrate the constructed graph on graph-convolutional recurrent neural networks and graph-convolutional transformers. Our model surpasses other baselines and achieves state-of-the-art performance on conventional traffic datasets. We demonstrate the superiority of our constructed graph by comparing its impact on other spatiotemporal models. We also show that our sensor and activity embedding approach can be easily scaled to other models, including pure temporal models like LSTM[14] and Transformer[29].

We summarize our contributions as follows:

- Graph construction: We propose a novel method to construct adjacency matrices that accurately represent vehicle behavior on actual roads, improving the sensor network.
- Sensor heterogeneity handling: We address individual sensor spatial heterogeneity by employing a sensor-specific one-hot encoding, allowing the model to adapt to diverse sensor environments.
- Human activity inference: We introduce a human activity embedding to directly infer from human actions, enhancing the explainability and accuracy of the models.
- Integration and performance: Our approach seamlessly integrates into popular network architectures and achieves state-of-the-art performance on real-world datasets.
- We contribute to the research community by publicly releasing our code, dataset, and experiment logs.[1]

2 RELATED WORK

2.1 Traffic prediction

Recent advancements in traffic prediction have prominently relied on the remarkable performance of spatiotemporal neural networks. However, the effective utilization of sensor proximity matrices remains ongoing discussion. Baseline works which are DCRNN and STGCN [18, 31] computes sensor proximity matrix from distance with gaussian filter, and perform graph convolution with the temporal model to facilitate spatiotemporal learning. More recently, various researches have proposed a model architecture that self-train the sensor similarity during training, mainly based on attention mechanism. ASTGCNN [12] use spatial attention matrix to adjust the weights of adjacency matrix dynamically and Graph-WaveNet[30] learn adjacency matrix in an end-to-end manner. While GMAN[33] leverages spatiotemporal embedding based on Node2vec[10] for attention blocks, they still allow a model to train spatial similarity during the spatial attention. STEP[25] leverages Gumbel-Softmax based trainable adjacency matrix, similar to GTS[24], while leveraging pretrained k-nearest neighbors (kNN) based graph adjacency for additional graph loss.

2.2 Urban Activity

Activity has been one of the key topics in travel demand analysis. [21, 22]. Compared to the traditional trip-based travel demand models (OD-based model), activity-based models are based on the principle that travel demand (trip-chain of ODs) is derived from people's daily activity patterns [4]. This type of models portray how people plan and schedule their daily travel [8] . These information lie decision making process of actual travelers and thus the sequence of urban activity should be found to understand travel behaviors.

While travel forecasting models based on trips have been widely employed, there are many criticisms that their lack of attention to the travel activities which reflect spatial and temporal consequences [17, 19, 26]. One of the dominant ideas underlying the activity-based approach is that human travels are faced with the inseparability and scarce nature of space and time [3, 20]. Activity-based models are becoming popular in transportation as it allows to infer the consequences of travel behavior that could not have been understood in trip-based models [16].

3 PROBLEM FORMULATION

We begin by formally defining the problem of spatiotemporal traffic prediction. We introduce a traffic sensor similarity graph $\mathcal{G} = (V, E, \mathbf{A})$, where V represents the set of sensors, E denotes the set of edges representing sensor similarity, and $\mathbf{A}$ represents the adjacency matrix. Hence, $N = |V|$ signifies the number of traffic sensors in the graph. At each time step t, the traffic values of the N sensors are represented by $X_t \in \mathbb{R}^N$. Additionally, we consider the frequency of urban human activity at time step t, denoted as $H_t \in \mathbb{R}^{K_H}$, where K_H indicates the number of categories for human activity.

Our problem is to learn a function f that predicts the next Q timesteps of traffic values, given a historical sequence of P timesteps of traffic values and $P + Q$ timesteps of estimated human activity frequencies $H_{t-P+1,\ldots,t+Q}$.

$$[X_{t-P+1}, \ldots, X_t; \mathcal{G}, H_{t-P+1,\ldots,t+Q}] \xrightarrow{f} X_{t+1}, \ldots, X_{t+Q} \quad (1)$$

[1] https://anonymous.4open.science/r/Traffic-UAGCRNTF-2BD4

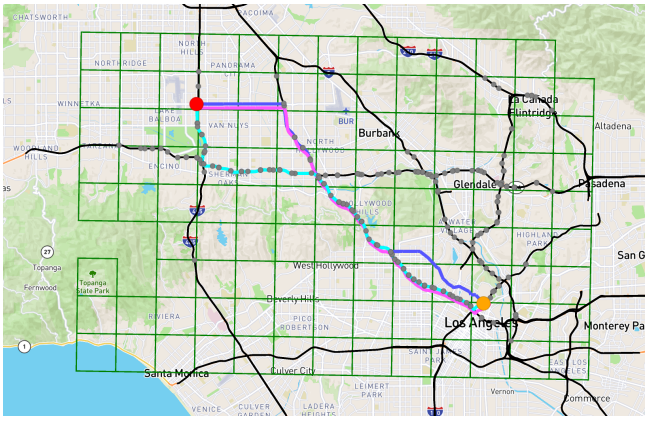


Figure 2: The A* algorithm is utilized to generate travel paths between the origin (red) and the destination (orange), sampled from a grid pair, with different costs of using the freeway: the ideal shortest path (blue), and paths that make greater use of the freeway (pink, cyan). The gray markers represent the traffic sensors, which do not necessarily appear on the generated travel paths (In METR-LA dataset).

4 METHODOLOGY

In this section, we present the methodology employed to address the traffic prediction problem. Our approach is composed of three key contributions: refined proximity graph construction, spatial sensor embedding, and urban activity embedding. These contributions form the foundation for our model architectures, namely UA-GCRN and UA-GCTransformer, which incorporate the Graph Convolutional Recurrent Network (GCRN) and attention-based Graph Transformer, respectively, as illustrated in Fig.3 and4. The term "UA" denotes the combination of sensor embedding (SE) and activity embedding (AE) added to the input of the encoder and decoder, to distinguish our application.

4.1 Graph Construction

4.1.1 Travel path generation. We partition the region in which traffic sensors are located into a grid of size $N_H^{(\text{Grid})} \times N_W^{(\text{Grid})}$. To generate plausible paths for each pair of grid regions, we employ the A* algorithm [13], resulting in a set of paths, $\mathcal{M}^{(Gen)}$. The A* algorithm efficiently explores the search space by combining uniform cost search and greedy best-first search. It selects the edge with the minimum cost as it progresses. In the A* algorithm, the cost of the next step movement is determined by the distance of the road, enabling the identification of the shortest path. By applying a coefficient to the cost of freeways during path generation, less than 1, we can obtain multiple paths with varying levels of freeway usage (see Fig.2).

4.1.2 Sensor Adjacency Matrix Construction. Firstly, we define the distance between two sensors, v_i and v_j, taking into account the road direction that can be traveled by car. Here, $i, j \in 1, ..., N$ represent the sensor indices. In our research, traffic sensors are installed on one-way freeways where sensors can be reached in consecutive sequence. As a result, the distance matrix is directed, implying that $\text{dist}(v_i, v_j) \neq \text{dist}(v_j, v_i)$.

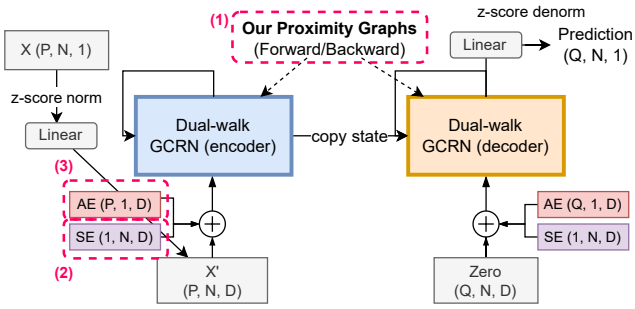


Figure 3: Model Architecture (UA-GCRN)

To construct the similarity matrix, we apply a Gaussian filter to the distance values. The distance-based proximity matrix between sensors v_i and v_j is computed as $A_{ij}^{(D)} = \exp\left(-\frac{\text{dist}(v_i,v_j)^2}{\sigma^2}\right)$ if $\text{dist}(v_i, v_j) < \kappa$ else 0. While previous works [18, 31] leveraged standard deviation [2] for σ and encountered ambiguity in selecting κ, we consider the specific characteristics of the traffic data. Taking into account the average traffic speed of approximately 60 mph (see Tab. 1) and an average distance traveled of 5 miles every 5 minutes (1 time-step), we set σ to 5 miles. Furthermore, we choose κ to be 80 miles, representing the maximum distance that can be covered within one hour (12 time-steps), which aligns with the observed maximum speed.

Next, to incorporate generated travel trajectories, we calculate the co-occurrence similarity[27] matrix $A_{ij}^{(S)}$, which measures the likelihood of paths between sensor nodes as follows:

$$A_{ij}^{(S)} = \frac{\#\text{ paths } v_i, v_j \text{ co-appear in } \mathcal{M}^{(Gen)}}{\sqrt{\#\text{ paths } v_i \text{ appears} \times \#\text{ paths } v_j \text{ appears in } \mathcal{M}^{(Gen)}}} \quad (2)$$

To obtain the final adjacency matrix to be used for graph convolution, we apply element-wise multiplication of the distance matrix and the co-occurrence matrix as $\mathbf{A} = \mathbf{A}^{(D)} \odot \mathbf{A}^{(S)}$.

4.2 Sensor Embedding

Each traffic sensor is situated within a unique built environment, resulting in distinct meanings in the actual traffic speed values. However, obtaining reliable sensor metadata to understand these variations is currently limited. To overcome this challenge, we adopt a similar strategy as described in [12], which involves the use of D-dimensional sensor embeddings (SE) generated through one-hot encoding of the N sensors. By incorporating these sensor embeddings, into the input of the encoder and decoder of our models, we can account for the individual characteristics of each sensor.

4.3 Activity Embedding

Urban human activity, driven by diverse travel purposes, significantly contributes to traffic congestion [5, 6]. To capture the temporal variations in human activity, we construct the activity frequency

[2]The measured standard deviations of distances are 4.97 miles (METR-LA), 3.93 miles (PEMS-BAY), and 6.92 miles (PEMSD7) respectively. However, it is important to note that the specific σ value can significantly fluctuate depending on the measured distances between sensors, which can potentially challenge the previous approach.

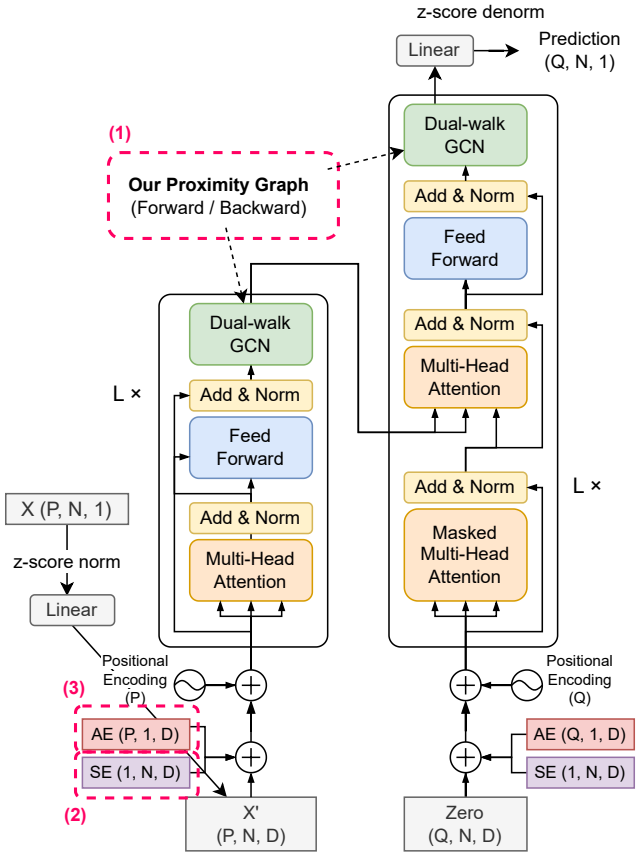


Figure 4: Model Architecture (UA-GCTransformer)

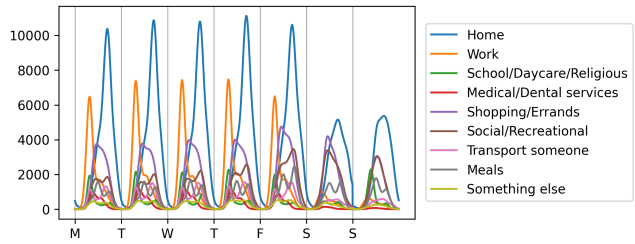


Figure 5: Urban human activity frequencies from the National Household Travel Survey for Activity Embedding.

based on a weekly pattern derived from the National Household Travel Survey [28], as depicted in Fig. 5. This allows us to create a representation $H_{t-P+1,\ldots,t+Q} \in \mathbb{R}^{K_H}$ that captures the estimated human activities for households at timestamp $t-P+1, \ldots, t+Q$. Subsequently, we first normalize the activity frequency with standard deviation, and is transformed into a D-dimensional activity embedding using a two-stacked dense layer followed by a normalization layer. To incorporate activity embedding into our models, we including $\mathrm{AE}_{t-P+1,\ldots,t}$ to the input for the encoder, and $\mathrm{AE}_{t+1,\ldots,t+Q}$ to the input for the decoder by addition, along with sensor embeddings. This allows our models to leverage the contextual activity information in both the encoding and decoding stages.

4.4 Deep Neural Network

In this section, we present UA-GCRN and UA-GCTransformer as our fundamental models that embody our proposed approach. However, variations of our models, such as UA-LSTM and UA-Transformer, can be implemented without utilizing graph convolution while still considering sensor and activity embedding. To leverage the constructed graph, we introduce dual-walk graph convolution. Additionally, we explore the application of dual-walk graph convolution in two different temporal deep learning methods: recurrent neural network (RNN) and Transformer.

4.4.1 Dual-walk Graph Convolution. We utilize a dual-walk graph convolution approach that combines both diffusion and reverse processes. This involves performing a multi-graph convolution using forward walk, backward walk, and the identity matrix. The motivation behind employing dual-walk convolution is to address traffic congestion that can occur in both directions due to traffic waves [9]. The dual-walk graph convolution is equivalent to a single-step dual-walk diffusion convolution, which was initially proposed in [18], expressed as follows:

$$g_{\theta\mathcal{G}} Z^{(l+1)} = [\theta_1(\mathbf{D}_{out}^{-1}\mathbf{A}) + \theta_2(\mathbf{D}_{in}^{-1}\mathbf{A}^T) + \theta_0(I)]Z^{(l)} \tag{3}$$

Here, $\theta_0, \theta_1, \theta_2$ represent trainable variables, and $\mathbf{D}_{out}$ and $\mathbf{D}_{in}$ are out-degree and in-degree diagonal matrices of $\mathbf{A}$, respectively. Additionally, we can efficiently perform sparse-matrix computations as our adjacency matrix is sparse.

4.4.2 Graph Convolutional Recurrent Network. We apply dual-walk graph convolutional on GCRN [18, 23] as illustrated in Fig. 3. Our UAGCRN module is equivalent to single-step dual-walk diffusion of DCRNN with sensor and activity embedding. The reason for employing a single-step instead of a multi-step approach, is due to the incorporation of well-engineered sensor connectivity within our graph structure, rendering the multiple diffusion unnecessary. This is experimentally demonstrated in Sec. 6.2.2. Moreover, a single GCRN module still can accumulate graph convolution of each time-step to enable multi-step prediction.

4.4.3 Graph Convolutional Transformer. The Transformer architecture [29] has achieved remarkable performance in language modeling tasks, leading to various attempts to adapt its structure for other domains. However, recent studies have relied on learnable positional encoding or learnable graph computation techniques [7, 15, 25, 33]. Interestingly, no existing model has successfully demonstrated the effectiveness of a basic Transformer on traffic prediction, without graph self-learning or modification in positional encodings. In our UA-GCTransformer model, we adopt the original Transformer architecture and utilize sinusoidal positional encoding to distinguish input and output sequence orders as Fig. 4. Additionally, we incorporate dual-walk graph convolution in each encoder and decoder layer, similarly in [12]. This approach explores the potential of language modeling while leveraging the power of graph convolution.

Table 1: Data statistics (B.C.: Normalized Betweenness Centrality). *PEMSD7 only contains weekdays.

	METR-LA	PEMS-BAY	PEMSD7*
# sensors (N)	207	325	228
Mean (mph)	54 (±20)	62 (±10)	59 (±13)
Data size	34,249	52,093	12,652
Start time	Mar/1/2012	Jan/1/2017	May/1/2012
End time	Jun/30/2012	May/31/2017	June/30/2012
# OSM roads	75,046	36,987	122,201
$N_H^{(\text{Grid})} \times N_W^{(\text{Grid})}$	9×13 (2mi.)	9×9 (2mi.)	8×12 (3mi.)
$\lvert\mathcal{M}^{(Gen)}\rvert$	105,361	46,205	66,510
Legacy Adj. NNZ	1,722 (4.0%)	2,694 (2.6%)	8,100 (15.6%)
Our Adj. NNZ	8,575 (20.%)	12,628 (12.0%)	7,135 (13.7%)
Mean B.C. Ours	3.04×10^{-3}	2.48×10^{-3}	3.32×10^{-3}

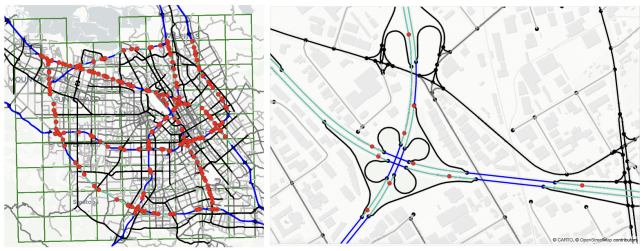


Figure 6: Traffic sensors (red markers) along with OSM freeways (blue paths) and the corresponding OSM freeways where the traffic sensors are located (green paths). The partitioned grid is also represented with dark green squares.

5 EXPERIMENTAL SETTING

5.1 Data Description and Preprocessing

We provide a description of the datasets used in our study and the corresponding preprocessing steps. Table 1 presents the statistics of datasets, including the number of nonzero weights (NNZ) in the adjacency matrix, and the mean betweenness centrality of our adjacency graph.

5.1.1 Traffic Datasets. We utilize three well-known traffic datasets: METR-LA [18], PEMS-BAY [18], and PEMSD7 [31]. These datasets contain information about traffic speeds recorded by sensors, as well as the original sensor adjacency matrix provided by the authors.

5.1.2 Open Street Map (OSM) Dataset. To accurately match the sensor locations to the corresponding roads, we leverage Open Street Map[2] data for the regions covered by the METR-LA, PEMS-BAY, and PEMSD7 datasets. We observed that the locations of some sensors do not align precisely with the OSM roads. In such cases, we updated the latitude, longitude, and freeway details of those sensors using the Caltrans Performance Measurement System (PeMS) [1].

5.1.3 Urban Activity Dataset. To incorporate urban human activity information, we extracted data from the National Household Travel Survey [28]. This dataset contains 828,438 travel surveys that include information about travel start and end times, as well as the mode of transportation (including car usage). We constructed an activity frequency histogram with a 5-minute resolution and smoothed pattern with gaussian filter (sigma=2), and used it as an input for activity embedding in our model. The number of activity categories is $K_H = 9$ and described in Fig. 5.

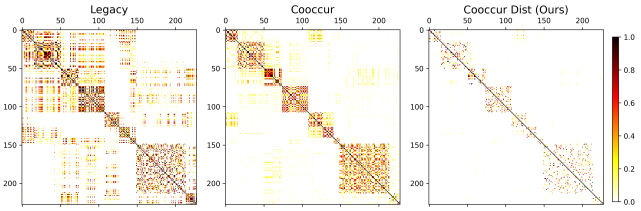


Figure 7: Adjacency Matrix Visualization: Legacy, Co-Occurrence Similarity, and Final Graph (PEMSD7).

5.1.4 Travel Path Generation. We conduct the travel path generation as illustrated in Sec. 4.1.1. We partition the area around the sensors into square grids of 2-3 miles, including padding, resulting in $N_H^{(\text{Grid})} \times N_W^{(\text{Grid})}$ grids. For $(N_H^{(\text{Grid})} \times N_W^{(\text{Grid})})^2$ pairs of grids, we attempt to generate a travel path using the A* algorithm. We perform this process 5 times with 3 different freeway costs (1.0, 0.9, 0.8 multiplied to freeway road length) for each grid pair as described in Fig. 2, resulting in a maximum of 15 roads being created[3]. As a result, we can generate $\mathcal{M}^{(Gen)}$ travel paths to construct our co-occurrence and distance-based adjacency matrix. As a example, adjacency matrix of PEMSD7 is visualized in Fig. 7.

5.2 Evaluation Setup

In our experiments, we evaluate MAE (Mean Absolute Error), RMSE (Root Mean Square Error), and MAPE (Mean Absolute Percentage Error) at 3, 6, and last (12 in METR-LA, PEMS-BAY, 9 in PEMSD7) step of prediction.

We utilized the following parameter settings: a batch size of 32, a hidden embedding dimension of 64, and the Adam optimizer with an initial learning rate of 0.01. We employed a patience of 5 for early stopping and reduced the learning rate to 1/10 after 2 trials. For the Transformer models, we employed 8 attention heads, a key dimension of 8, a total dimension of 64, and stacked 3 layers.

6 RESULTS

6.1 Performance Comparison

We have selected several baselines for comparison with our proposed UAGCRN, UAGCTransformer. The baselines include the following models[4]:

- **Last Repeat**: This baseline simply repeats the last observed sensor readings as predictions.
- **LSTM**: Long Short-Term Memory is commonly used for sequential data processing.

[3]Note that there can be cases where a path is not established.
[4]Although, we considered Traffic Transfomer[7] for its state-of-the-art performance, we encountered difficulties in finding a reliable dataset or test logs. We assume there might be confusion in metrics such as averaging over multiple time steps.

Table 2: Forecasting error in METR-LA, PEMS-BAY, PEMSD7 datasets. † represents the model leveraging our co-occurence and distance based adjacency matrix. * represents the model self trains the sensor similarity. Best and second best results are represented as BOLD and underline.

			Temporal Only					Spatiotemporal								
		Metric	LastRepeat	LSTM	TF	UA-LSTM	UA-TF	DCRNN	DCRNN†	GTS*[25]	STGCN	GWNet*	GMAN*	STEP*	**UAGCRN†**	UAGCTF†
METR-LA	15 min	MAE	4.02	3.09	3.07	2.82	2.81	2.77	2.67	2.67	2.88	2.69	2.77	**2.61**	2.64	2.63
		RMSE	8.69	6.10	6.09	5.56	5.58	5.38	5.21	5.27	5.74	5.15	5.48	**4.98**	5.09	5.07
		MAPE	9.4%	8.2%	8.1%	7.5%	7.6%	7.30%	6.89%	7.21%	7.62%	6.90%	7.25%	**6.60%**	6.77%	6.71%
	30 min	MAE	5.09	3.79	3.76	3.19	3.18	3.15	3.06	3.04	3.47	3.07	3.07	**2.96**	2.97	2.96
		RMSE	11.13	7.66	7.65	6.59	6.61	6.45	6.30	6.25	7.24	6.22	6.34	**5.97**	6.08	6.04
		MAPE	12.2%	10.7%	10.6%	9.1%	9.1%	8.80%	8.38%	8.41%	9.57%	8.37%	8.35%	**7.96%**	8.10%	8.08%
	60 min	MAE	6.80	4.90	4.88	3.56	3.54	3.60	3.56	3.46	4.59	3.53	3.40	3.37	3.35	**3.34**
		RMSE	14.21	9.68	9.67	7.55	7.52	7.59	7.52	7.31	9.40	7.37	7.21	**6.99**	7.12	7.02
		MAPE	16.7%	14.9%	14.8%	10.6%	10.7%	10.50%	10.15%	9.98%	12.70%	10.01%	9.72%	**9.61%**	9.68%	9.65%
PEMS-BAY	15 min	MAE	1.60	1.45	1.45	1.32	1.33	1.38	1.29	1.34	1.36	1.30	1.34	**1.26**	1.30	1.30
		RMSE	3.43	3.16	3.16	2.82	2.85	2.95	**2.72**	2.83	2.96	2.74	2.82	2.73	2.73	2.76
		MAPE	3.2%	3.0%	3.0%	2.8%	2.8%	2.90%	2.69%	2.82%	2.90%	2.73%	2.81%	**2.59%**	2.71%	2.75%
	30 min	MAE	2.18	1.98	1.98	1.63	1.63	1.74	1.62	1.66	1.81	1.63	1.62	**1.55**	1.61	1.61
		RMSE	4.99	4.61	4.61	3.77	3.78	3.97	3.68	3.78	4.27	3.70	3.72	**3.58**	3.68	3.70
		MAPE	4.7%	4.5%	4.5%	3.7%	3.7%	3.90%	3.62%	3.77%	4.17%	3.67%	3.63%	**3.43%**	3.62%	3.64%
	60 min	MAE	3.05	2.72	2.71	1.89	1.88	2.07	1.92	1.95	2.49	1.95	1.86	**1.79**	1.87	1.86
		RMSE	7.01	6.28	6.27	4.41	4.40	4.74	4.45	4.43	5.69	4.52	4.32	**4.20**	4.37	4.33
		MAPE	6.8%	6.8%	6.7%	4.5%	4.4%	4.90%	4.52%	4.58%	5.79%	4.63%	4.31%	**4.18%**	4.39%	4.36%
PEMSD7	15 min	MAE	2.49	2.35	2.37	2.13	2.13	2.21	2.10	2.21	2.25	2.31	2.30	2.09	**2.05**	2.06
		RMSE	4.65	4.48	4.51	4.03	4.12	4.21	3.98	4.16	4.04	4.44	4.39	3.99	**3.87**	3.93
		MAPE	5.7%	5.5%	5.5%	5.1%	5.1%	5.14%	4.91%	5.15%	5.26%	5.41%	5.66%	5.00%	**4.85%**	4.86%
	30 min	MAE	3.51	3.31	3.33	2.71	2.69	3.01	2.75	2.95	3.03	3.26	2.71	2.66	2.61	**2.59**
		RMSE	6.77	6.49	6.53	5.37	5.49	5.96	5.45	5.74	5.70	6.41	5.35	5.37	**5.20**	5.22
		MAPE	8.3%	8.1%	8.1%	6.9%	6.9%	7.43%	6.85%	7.43%	7.33%	8.11%	6.87%	6.80%	6.56%	**6.50%**
	45 min	MAE	4.31	4.05	4.10	3.01	2.98	3.59	3.19	3.47	3.57	4.63	2.99	2.95	2.92	**2.90**
		RMSE	8.32	7.89	7.99	6.10	6.13	7.14	6.39	6.78	6.77	8.81	5.94	6.03	**5.90**	5.91
		MAPE	10.4%	10.3%	10.3%	7.9%	7.8%	9.18%	8.24%	9.06%	8.69%	12.40%	7.70%	7.74%	7.58%	**7.48%**

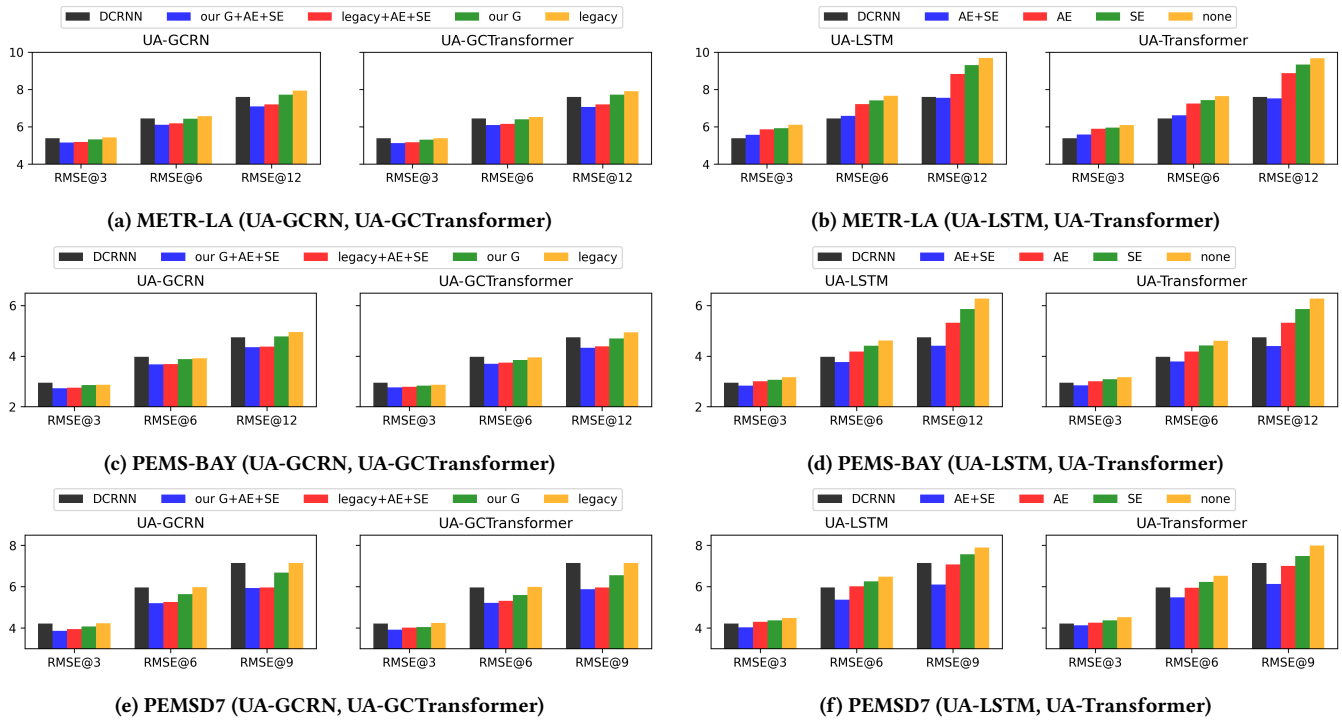
- **Transformer**[29][5]: Transformer is well known for its success in natural language processing tasks.
- **DCRNN** [18]: This model is based on the GCRN architecture and is designed specifically for traffic forecasting.
- **GTS** [24]: GTS shares the same architecture as DCRNN, but it trains a probable graph similarity between sensor pairs.
- **STGCN** [31]: STGCN utilizes graph convolutions with the Laplacian matrix to capture spatial dependencies.
- **Graph WaveNet (GWNet)** [30]: GWNet applies a self-adaptive adjacency matrix in an end-to-end manner.
- **GMAN** [33]: GMAN leverages Node2vec trained from an adjacency matrix, but it also conducts spatial attention, allowing every pair of sensors to have similarity.
- **STEP** [25]: STEP is the state-of-the-art model which adopts a masked encoder-decoder. It preprocesses the sensor similarity based on the k-NN algorithm.

Additionally, we also compare our proposed models with two variants: UA-LSTM and UA-Transformer. These variants leverage activity and sensor embeddings but do not incorporate graph convolutions, enabling us to evaluate the impact of graph utilization.

6.1.1 Forecasting error. Tab. 2 presents the results of comparing various baseline models with our proposed models (UAGCRN and UAGCTransformer). On the header of table, † represents the model leveraging our co-occurence and distance based adjacency matrix, and * represents the model self-learns the sensor similarity, which are GTS, GWNet, GMAN, and STEP.

Overall, the proposed models, UAGCRN and UAGCTransformer, consistently outperform the major spatiotemporal baselines across all three datasets and various time intervals. Moreover, these models

[5]We followed the TensorFlow official tutorial for the Transformer model, with the word embedding layer replaced: https://www.tensorflow.org/text/tutorials/transformer.



(a) METR-LA (UA-GCRN, UA-GCTransformer)

(b) METR-LA (UA-LSTM, UA-Transformer)

(c) PEMS-BAY (UA-GCRN, UA-GCTransformer)

(d) PEMS-BAY (UA-LSTM, UA-Transformer)

(e) PEMSD7 (UA-GCRN, UA-GCTransformer)

(f) PEMSD7 (UA-LSTM, UA-Transformer)

Figure 8: Ablation Test (RMSE) of our modules – Our Graph(G), SE, AE.

outperforms UALSTM and UATransformer, indicating that incorporating graph convolutions significantly improve the accuracy of traffic forecasting models.

We further evaluate the effectiveness of our graph construction approach by comparing DCRNN, DCRNN†, and GTS[6], which share the same architecture. Notably, we observed that DCRNN† outperforms GTS, particularly on the PEMS-BAY and PEMSD7 datasets. The superior performance of DCRNN† can be attributed to the fact that GTS considers all potential sensor connections, often resulting in biased predictions. We also noticed this issue of trainable graph similarity in GWNet, as it exhibites significant errors on the PEMSD7 dataset. The sensor networks in the PEMSD7 dataset have higher betweenness centrality (Tab. 1), indicating that the graph structure provides more valuable information compared to other datasets. These complexities in sensor similarity may also contribute to the lower performance of the STEP model compared to our approach, as STEP relies on a data-driven approach that may not accurately capture the intricate sensor relationships.

Moreover, our proposed UA approach, which includes SE and AE components, significantly improves the performance of purely temporal models (LSTM and TF). UA-LSTM and UA-Transformer surpasses other spatiotemporal baselines such as DCRNN, GTS, STGCN, GWNet, and GMAN on the PEMS-BAY and PEMSD7 datasets. This observation suggests that by incorporating $(P + Q) \times N$ types of inputs, which include both sensor index and urban activity context, our models are able to distinguish between different sensor inputs and capture distinct activity contexts. This comprehensive input representation empowers our models to generate accurate predictions for multi-step traffic forecasting tasks.

However, we observe limited improvement of Transformer over LSTM and UAGCTransformer over UAGCRN. This can be attributed to the fact that traffic prediction involves relatively short time-series steps, unlike in the case of large language models, where Transformers excel. Consequently, RNN models continue to perform well in this domain.

Although our proposed models, UAGCRN† and UAGCTF†, are outperformed by the more complex model STEP on the METR-LA and PEMS-BAY datasets, STEP utilizes very long patches of input (e.g. $P = 228 \times 7$) which allows it to capture more complex and intricate patterns. This approach results in a heavier model that can handle larger contextual information, as it shows better performance in longer timesteps. Despite the performance difference, our models still demonstrate potential for improvement, will be explained in Sec. 6.2.3. On the other hand, we believe that studying STEP's architecture and mechanisms can provide valuable insights to advance the state-of-the-art in related models.

6.1.2 Computational Cost. We conducted a comparison of the computational cost for each model in their default settings in Tab. 3[7]. In order to ensure a fair comparison we leverage default settings of each model such as DCRNN with 3 diffusion steps, GMAN with $L = 5$. We were unable to precisely measure the computational cost of STEP[25] under the same environment. However, during our

[6]Results from [25] due to issues with GTS: https://github.com/chaoshangcs/GTS/issues

[7]Tab. 2 is based solely on the original author's implementation, while Tab. 3 is intended for evaluating computational time under same learning framework (TensorFlow2) and GPU (RTX3090), batch size, and early stopping condition.

experiments of STEP with the PEMSD7 dataset (2.7 times smaller than METR-LA), each epoch took approximately 3.5 minutes to train using 3 TITAN RTX GPUs. The total training time was approximately 5 hours. The result shows that UAGCRN outperforms other models in terms of computational cost and training time.

Table 3: Computational cost of METR-LA under same environment. The number of stacks are $L = 5$ in GMAN and $L = 3$ in UAGCTF†, while DCRNN, UAGCRN† do not have stacked architecture ($L = 1$).

	DCRNN	GMAN	UAGCRN†	UAGCTF†
# Params	353,025	714,049	**174,401**	842,177
Train (m:s/ep.)	2:35	4:39	**42s**	4:26
Total Epochs	26	19	24	17
Total train time	1:12:03	1:34:41	**0:18:36**	1:21:04

6.2 Ablation Study

6.2.1 Effectness of individual module (Graph, AE, SE). The results of the ablation test for each module are presented in Fig. 8. Specifically, Fig. 8a,8c,8e demonstrate the performance improvement of UA-GCRN and UA-GCTransformer, when using our graph compared to the legacy graph. These results indicate that our graph contains more traffic-related knowledge regarding sensor correlation. Although the enhancement looks marginal when both the SE and AE modules are given under the same conditions, it still highlights the potential for performance improvement in less common situations.

Furthermore, Fig. 8b,8c,8f showcase the effectiveness of each SE and AE module on temporal-only models, specifically UA-LSTM and UA-Transformer. The performance improves as these modules are integrated. Notably, the impact of the AE module is more significant than the SE module, likely due to traffic patterns exhibiting a stronger correlation with human activity. Additionally, incorporating the SE module to account for sensor spatial heterogeneity further enhances performance. When both SE and AE modules are integrated, the resulting performance surpasses that of DCRNN in the PEMS-BAY and PEMSD7 datasets.

6.2.2 The number of diffusion steps of DCRNN with our graph. Figure 9 depicts the results of UADCGRU† obtained by applying the SE and AE to the DCGRU model using our graph while modifying the diffusion steps. In contrast to the original findings discussed in [18] which suggested a demand for approximately 3 diffusion steps, our results show that increasing the graph connectivity information, along with activity and sensor data, can lead to worse performance. We analyze that vehicles do not follow a random-walk pattern, and the vehicle travel pattern is already adequately captured in our constructed graph.

6.2.3 Comparison of Timestamp Embedding and Activity Embedding. Various models have employed different approaches to incorporate timestamp information. For example, in DCRNN, the time of day is included as an additional input channel[8]. In this ablation

[8]Not mentioned in the paper, but in the code: https://github.com/liyaguang/DCRNN

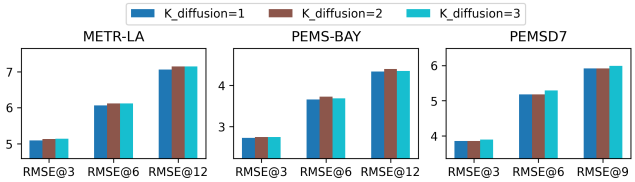


Figure 9: Performance degradation in UADCGRU† as the number of diffusion steps (K) increases.

study, we compared timestamp embedding (TE), which are generated from a vector space $\{0, 1\}^{7+12\times24}$ (one-hot concatenation of weekday and time-of-day) and ingested in 2-stacked dense layer with normalization layer to capture weekly and daily periodicity, similar to [15, 33], with AE.

Tab. 4 shows the comparison results of UAGCRN† with TE or AE, indicating that TE exhibited a slight improvement over AE, performing almost as well as STEP in the METR-LA and PEMS-BAY datasets, and outperforming AE in the PEMSD7 dataset. The performance improvement of the TE over the AE can be attributed to the lack of analysis in localized activity patterns of each city when we estimate human activity frequency which is derived from national surveys. This aspect suggests that future studies should consider accurately inputting AE, such as localized activity estimation considering demographics and urban function.

On the other hand, relying on one-hot timestamp information results in less explainability due to its discrete nature, unlike the continuous activity information. Additionally, it may pose limitations in scalability when accounting for seasonal effects in long-term datasets, while our datasets are deal with only a few months (Tab. 1). Nevertheless, we can still take advantage of the AE-based UAGCRN model for its superior explainability.

6.3 Case Study

Fig.10a and Fig.10b present a case study illustrating the superior performance of UA-GCRN† with our graph, sensor and activity embeddings. In both cases, the legacy graph includes incorrect connections that cannot be reached from the target sensor, which cause in wrong prediction.

In the METR-LA dataset (Fig. 10a), UA-GCRN† achieves better congestion prediction even without sensor and activity embeddings by accurately establishing connections between roads. This highlights the effectiveness of our approach in constructing the graph, which significantly improves the model's performance.

Furthermore, in the PEMS-BAY dataset (Fig. 10b), we observe that UA-GCRN† performs even better when provided with activity input. In this case, a high frequency of work activity is included in the historical sequence, and possible shopping activity in future sequence, which helps the model there can be consequent congestion in the prediction steps. The results demonstrate the additional benefit of incorporating activity information into the model, further enhancing its performance.

Table 4: Ablation study of UAGCRN† and UAGCTF† by replacing AE with timestamp embedding (TE). Best and second best results are represented as BOLD and underline.

		STEP	UAGCRN†		UAGCTF†	
			TE+SE	AE+SE	TE+SE	AE+SE
METR-LA	MAE3	**2.61**	2.62	2.64	2.63	2.63
	RMSE3	**4.98**	5.00	5.09	5.09	5.07
	MAE6	2.96	**2.94**	2.97	2.95	2.96
	RMSE6	5.97	**5.97**	6.08	6.05	6.04
	MAEL	3.37	**3.31**	3.35	3.35	3.34
	RMSEL	**6.99**	7.02	7.12	7.10	7.02
PEMS-BAY	MAE3	**1.26**	1.28	1.30	1.28	1.30
	RMSE3	2.73	**2.69**	2.73	2.72	2.76
	MAE6	**1.55**	1.60	1.61	1.59	1.61
	RMSE6	**3.58**	3.63	3.68	3.66	3.70
	MAEL	**1.79**	1.88	1.87	1.86	1.86
	RMSEL	**4.20**	4.38	4.37	4.37	4.33
PEMSD7	MAE3	2.09	**2.02**	2.05	2.04	2.06
	RMSE3	3.99	**3.81**	3.87	3.88	3.93
	MAE6	2.66	**2.56**	2.61	2.57	2.59
	RMSE6	5.37	**5.14**	5.20	5.16	5.22
	MAEL	2.95	**2.88**	2.92	2.89	2.90
	RMSEL	6.03	**5.88**	5.90	5.88	5.91

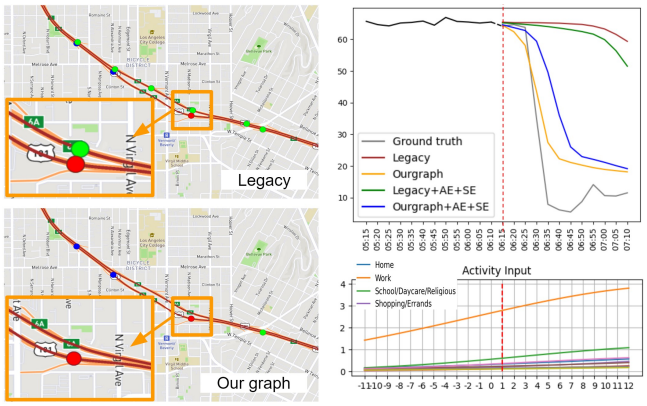
6.4 Analysis of Sensor Reactions Based on Activity Information

We examine how sensors react differently when provided with different activity information. We conducted tests by setting all sensors to a speed value of 30 mph during the P sequence while varying the activity input, as illustrated in Fig. 11. The choice of 30 mph is for testing whether congestion would increase or alleviate when the road capacity is full. We conducted two predictions of next 15 min, pred1 and pred2, by providing activity information for the morning rush hour (6:35 to 8:20) and the evening work time (16:45 to 18:30), respectively.

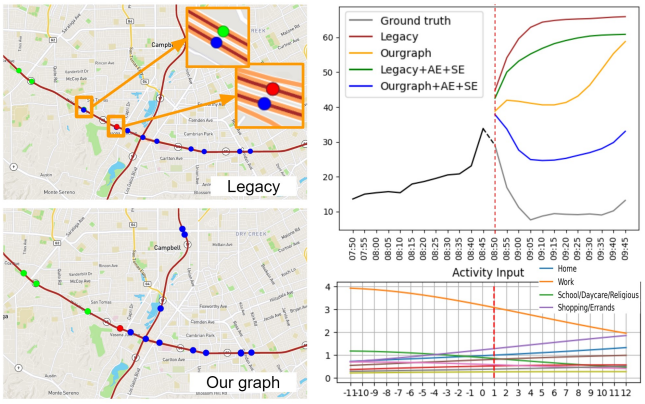
Our findings revealed that sensors exhibited different behaviors based on the given activities. This discrepancy is due to varying levels of road utilization associated with specific activities. Notably, even when the same traffic values are given to the model, our model predicted distinct patterns as it had learned the sensor's typical response patterns corresponding to future activities. Overall, these analyses highlight the importance of incorporating sensor embedding while inserting activity information into traffic prediction models, as it leads to a better understanding of sensor reactions and enhances the accuracy of congestion predictions influenced by urban human activity.

7 DISCUSSION

Our current approach focuses on activity-based traffic prediction models that incorporate travel purposes in space and time. However, there still a few improvements points. Firstly, to generate more realistic paths, it is important to consider travel demands and purposes between different areas, taking into account factors such as land use, urban function, and demographic information. Additionally, analyzing localized activities for each sensor can provide valuable insights into congestion patterns, which can be inferred from the built environment, including road networks and points



(a) METR-LA, outperforms with our graph (ID: 716339)



(b) PEMS-BAY, outperforms with our graph with AE (ID: 400688)

Figure 10: Case study: UAGCRN on METR-LA and PEMS-BAY. Activity labels follows Fig. 5. The target sensor (red), forward connected sensors (green), backward connected sensors (blue) are represented with colored markers on the map. Erroneous connections are found in legacy graphs.

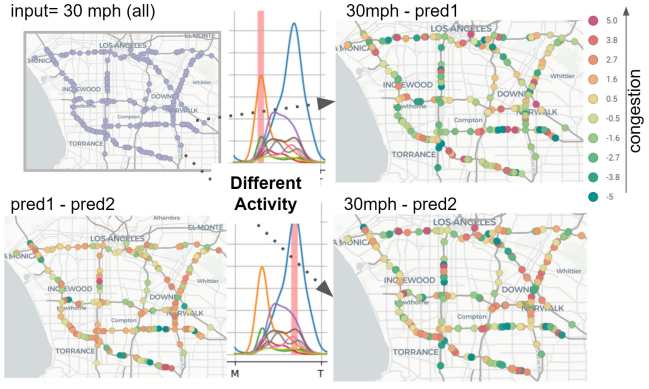


Figure 11: Sensor Reactions Based on Activity Information with UAGCRN (Red/Green: more/less congestion)

of interest (POIs). Furthermore, integrating real-time data sources, such as real-time transportation data or social media data, into a dynamic graph representation is crucial for capturing real-time variations in traffic and adapting to changing conditions. By refining our methodology in these areas, future researches can enhance the accuracy and applicability of our models, allowing us to better understand and predict traffic patterns in urban areas.

8 CONCLUSION

In conclusion, our research highlights the advantages of integrating real-world knowledge of urban human activity into spatiotemporal traffic prediction models. We propose a novel approach that effectively addresses the challenges of accurate graph construction, individual sensor heterogeneity handling, and human activity-based inference. Our proposed method incorporates realistic travel path generation with A* algorithm, co-occurrence and distance-based sensor connectivity measures, sensor-specific one-hot encoding, and human activity embedding into graph-convolution based spatiotemporal deep learning architectures. Experimental results demonstrate the effectiveness of our approach, surpassing other baselines and achieving state-of-the-art performance on real-world datasets. The insights gained from this study contribute to a better understanding of traffic patterns influenced by urban human activity, opening up avenues for further advancements in traffic prediction.

REFERENCES

[1] [n. d.]. California Performance Measurement System (PeMS). https://pems.dot.ca.gov/. Accessed: June 3, 2023.

[2] [n. d.]. OpenStreetMap. https://www.openstreetmap.org. Accessed: June 3, 2023.

[3] Kay W Axhausen and Tommy Gärling. 1992. Activity-based approaches to travel analysis: conceptual frameworks, models, and research problems. *Transport reviews* 12, 4 (1992), 323–341.

[4] Moshe E Ben-Akiva and John L Bowman. 1998. Activity based travel demand model systems. *Equilibrium and advanced transportation modelling* (1998), 27–46.

[5] Chandra R Bhat, Jessica Y Guo, Sivaramakrishnan Srinivasan, and Aruna Sivakumar. 2004. Comprehensive econometric microsimulator for daily activity-travel patterns. *Transportation Research Record* 1894, 1 (2004), 57–66.

[6] John L Bowman and Moshe E Ben-Akiva. 2001. Activity-based disaggregate travel demand model system with activity schedules. *Transportation research part a: policy and practice* 35, 1 (2001), 1–28.

[7] Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, and Rui Zhu. 2020. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS* 24, 3 (2020), 736–755.

[8] Joe Castiglione, Mark Bradley, and John Gliebe. 2015. *Activity-based travel demand models: a primer.* Number SHRP 2 Report S2-C46-RR-1.

[9] Carlos F Daganzo. 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation research part B: methodological* 28, 4 (1994), 269–287.

[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[11] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 922–929.

[12] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5415–5428.

[13] Peter E Hart, Nils J Nilsson, and Bertram Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4, 2 (1968), 100–107.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. *arXiv preprint arXiv:2301.07945* (2023).

[16] Johan W Joubert and Alta De Waal. 2020. Activity-based travel demand generation using Bayesian networks. *Transportation Research Part C: Emerging Technologies* 120 (2020), 102804.

[17] Ryuichi Kitamura. 1988. An evaluation of activity-based travel analysis. *Transportation* 15 (1988), 9–34.

[18] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJiHXGWAZ

[19] Michael G McNally. 2000. THE FOUR-STEP MODEL. IN: HANDBOOK OF TRANSPORT MODELLING. (2000).

[20] Tijs Neutens, Tim Schwanen, and Frank Witlox. 2011. The prism of everyday life: Towards a new research agenda for time geography. *Transport reviews* 31, 1 (2011), 25–47.

[21] Abdul Rawoof Pinjari and Chandra R Bhat. 2011. Activity-based travel demand analysis. In *A handbook of transport Economics*. Edward Elgar Publishing.

[22] Christian M Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C González. 2013. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface* 10, 84 (2013), 20130246.

[23] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*. Springer, 362–373.

[24] Chao Shang, Jie Chen, and Jinbo Bi. 2021. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=WEHSlH5mOk

[25] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. 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*. 1567–1577.

[26] Bruce D Spear. 1996. New approaches to transportation forecasting models: A synthesis of four research proposals. *Transportation* 23 (1996), 215–240.

[27] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* 3, Dec (2002), 583–617.

[28] Federal Highway Administration U.S. Department of Transportation. 2017. 2017 National Household Travel Survey. (2017). http://nhts.ornl.gov

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[30] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)* (2019).

[31] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3634–3640. https://doi.org/10.24963/ijcai.2018/505

[32] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (2019), 3848–3858.

[33] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 1234–1241.