Enhancing Spatiotemporal Traffic Prediction through Urban Human Activity Analysis

Sumin Han[1], Youngjun Park[1], Minji Lee[1],
Jisun An[2], Dongman Lee[1]

[1]School of Computing,
Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

[2]Luddy School of Informatics, Computing, and, Engineering,
Indiana University Bloomington (IUB), USA

32nd ACM International Conference on Information and Knowledge Management

October 21–25, 2023

KAIST

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2019-0-01126, Self-learning based Autonomic IoT Edge Computing).

Table of Contents

- Introduction
- Problem Formulation
- Methodology
- Experimental Results
- Discussion & Conclusion

Table of Contents

Introduction

- Traffic (speed/volume) prediction ensures safety and convenience.
- By leveraging historical data with **sensor adjacency network**, we can forecast future congestion.
- Prior research explored effective **spatiotemporal graph-based deep-learning models.**
  - Graph RNN: DCRNN (ICLR-18), STGCN (IJCAI-18), GTS (ICLR-21), ASTGCN (AAAI-19), T-GCN (TITS-19)
  - Graph Transformers: GMAN (AAAI-20), ASTGCNNTF (TKDE-22), STEP (KDD-22), PDFormer (AAAI-23)

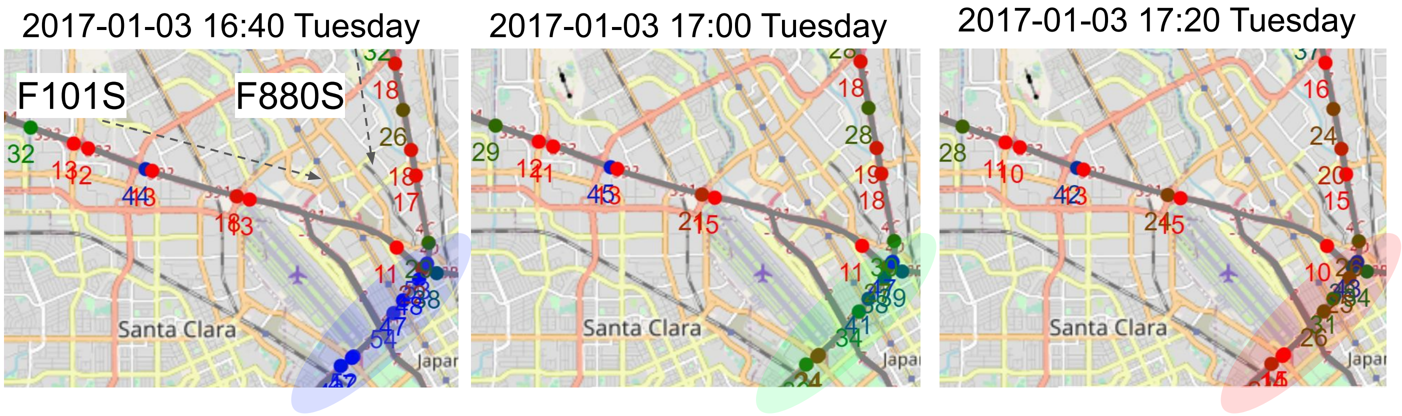


Fig. Congestion pattern of sensors in PEMS-BAY dataset (red > green > blue).

Challenges 1: Sensor adjacency matrix construction

- Deficiency in **justification for an adjacency matrix** construction (how far, how strong).
  - DCRNN, STGCN leverages distance-based Gaussian proximity as sensor adjacency matrix, but often **erroneous and unreasonable connections** are found.
  - Recent **data-driven approaches** with trainable graph adjacency are proposed (e.g. Gumbel Softmax, Graph attention), but they may generate **false connection** even there is no causal relationship.

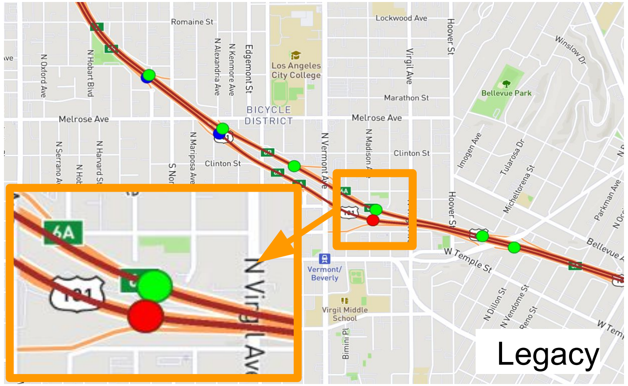


Fig. Erroneous connections are found in legacy graphs: The target sensor (red), forward (green), backward (blue).

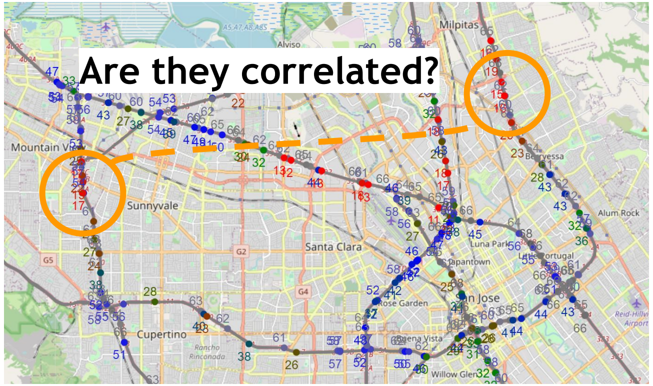


Fig. Problems with data-driven sensor adjacency construction (sensors with no causal relationship can be connected due to similar pattern in data).

Challenges 2: Sensor Heterogeneity

- **Sensor heterogeneity – meanings of each sensor value can be different.**
  - Different sensor position in lanes, road size (#lanes), intersections, road network, etc.
  - Some previous works (e.g. ASTGCNN) has addressed this issue by leveraging **spatial positional encoding**, however their main focus were on construction of the *positional encoding* for Transformer.

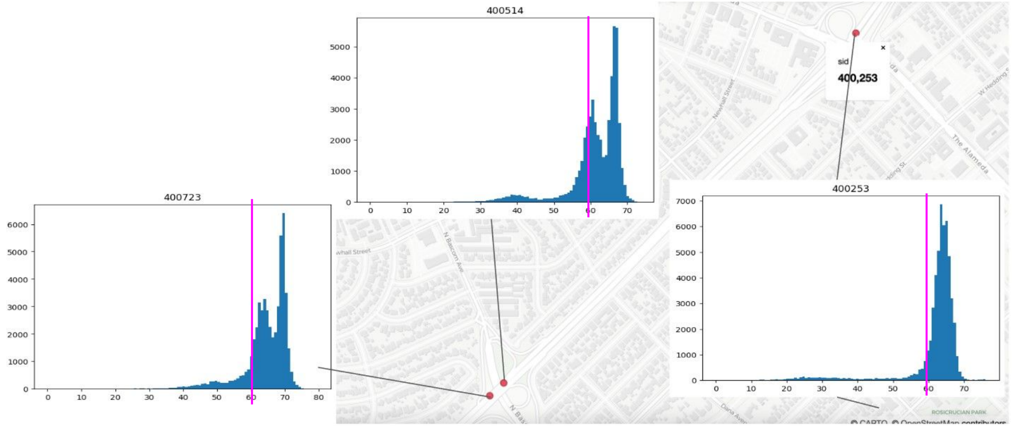


Fig. Speed histogram of each traffic sensors. Same 60mph can represent different meanings for each traffic sensors.

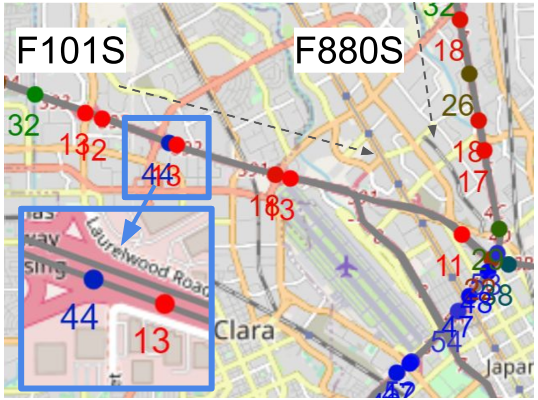


Fig. Both 44mph in sensor A and 13mph in sensor B can imply "congestion".

Challenges 3: Human Activity-based Prediction

- **Human activities**, such as commuting, significantly influence traffic patterns and can lead to congestion.
- Previous research only leverages discrete timestamp information (e.g. weekday, time of a day).
  - E.g. Monday ⇒ [1, 0, 0, …], Tuesday ⇒ [0, 1, 0, …], …
  - E.g. 0:00 ⇒ [1, 0, 0, …], 0:05 ⇒ [0, 1, 0, …], 0:10 ⇒ [0, 0, 1, …], …
- While temporal information provides insights into human activities, it does not establish direct causality.
  - Traffic patterns are caused by human actions.

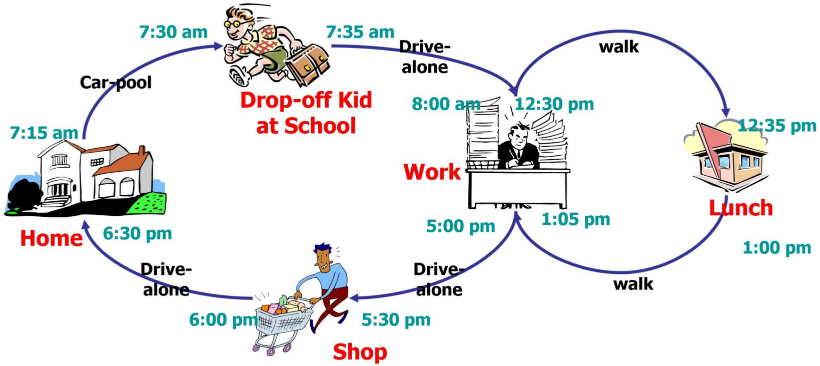
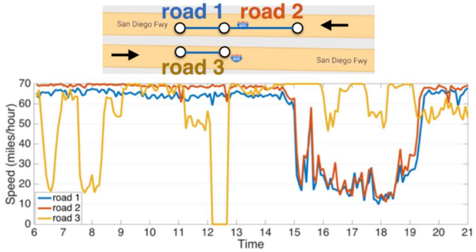


Fig. Daily travel pattern of a person based on human activity.

Problem Formulation

- Aims to predict traffic values (speed, volume) from recent history to forecast near future values, leveraging the **traffic sensor network**.
- Problem formulation:
  - **X_*t***: traffic values of ***N*** sensor at the timestamp ***t***.
  - ***G*:** sensor adjacency graph
  - **H_*t***: estimated human activity frequency at the timestamp ***t***.
  - Problem statement: find optimal model *f(·)* for seq-to-seq prediction,

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

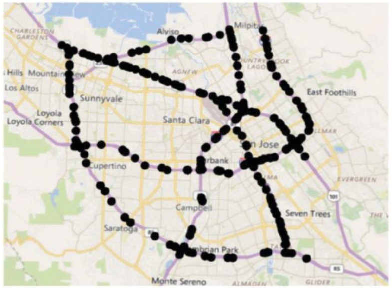


P-seq

Prediction f(·))

Q-seq

	sensor_0	sensor_1	sensor_2	sensor_n
2018/01/01 00:00:00	60.0	65.0	70.0	...
2018/01/01 00:05:00	61.0	64.0	65.0	...
2018/01/01 00:10:00	63.0	65.0	60.0	...
...	...	...	...	...



Road Network with Traffic Sensors

Table of Contents

Overall architecture of our model

- Our key contribution:
  - **(1) Graph construction, (2) Sensor embedding (SE), (3) Activity embedding (AE)**
- Overall dataflow: Input traffic values are z-score normalized (std, mean), and ingested to 2-stacked dense layers to make *D* dimensional embedding, and z-score denormalized for output.

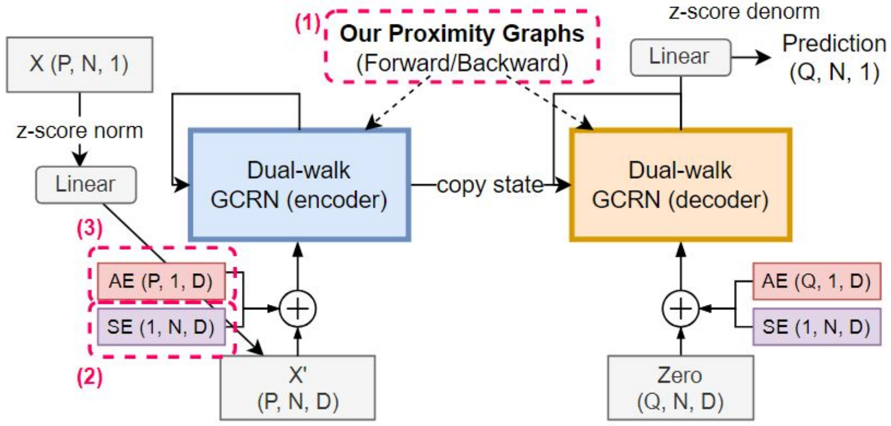


Figure 3: Model Architecture (UA-GCRN)

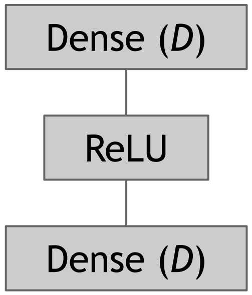


Fig. 2-stacked dense layers with ReLU activation.

(1) Graph Construction: Travel Path Generation

- We generate travel path using **A* shortest path algorithm** on the OpenStreetMap road network.
  - Vehicle movement follows shortest time consuming paths, not random walk.
- Make grids of 2~3 miles size ($N_H^{(\mathrm{Grid})} \times N_W^{(\mathrm{Grid})}$), and generate travel path for **each orient-destination grid pairs**.
  - Parameterize the cost of using freeway (1.0, 0.9, 0.8), generate multiple probable travel paths prefers freeway use.
  - METR-LA (105k), PEMS-BAY (46k), PEMSD7 (66k) travel paths are generated.

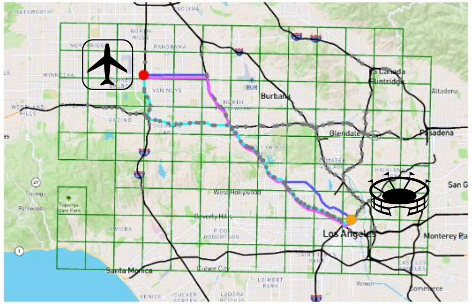


Fig. Split area into N_H × N_W grids (green squares). Orient to destination with different parameters of highway cost (blue < pink < cyan uses more freeway).

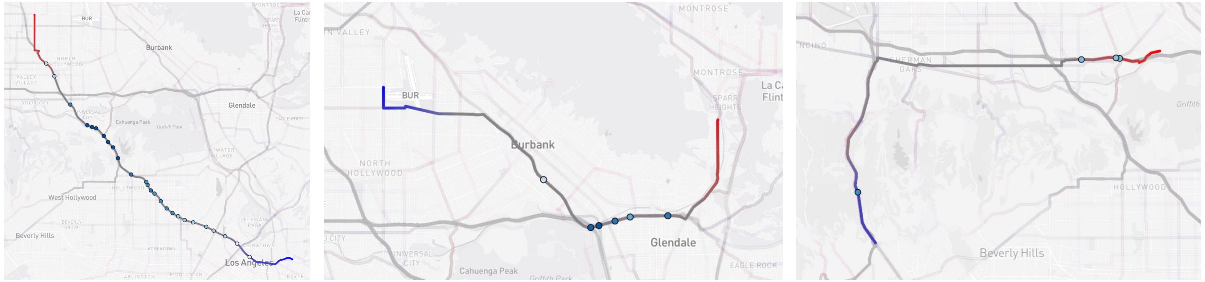


Fig. Samples of generated travel paths (orient: blue, destination: red).

(1) Graph Construction: Co-occurrence Matrix

- Treat the generated travel paths like sentences with words, and we calculate co-occurrence similarity (normalized mutual information).

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

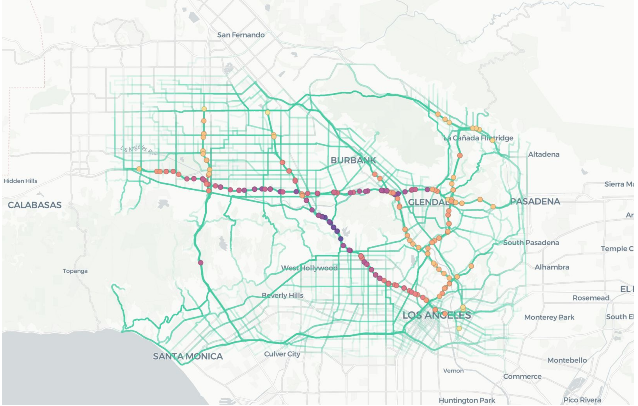


Fig. Stacked visualization of generated travel paths.
(darker color – frequency of the sensor appearance)

… '68218060', '26404567', **'312118657'**, **'S400479'**, '52140457', '4104395806', 'S400030', '4977823055', 'S401440', '245810051', '287826065', 'S403225', …

Fig. A travel path contains OSM node IDs and sensor IDs like a sentence.

(1) Graph Construction: Distance-based Proximity

- A traffic wave takes more time steps to reach distant sensors.
  - Construct distance-based Gaussian proximity matrix [DCRNN, STGCN]:

$$A_{ij}^{(D)} = \exp\left(-\frac{dist(v_i,v_j)^2}{\sigma^2}\right) \text{ if } dist(v_i,v_j) < \kappa \text{ else } 0.$$

  - We set σ = 5 miles and κ = 80 miles.
    - **Average traffic speed is around 60 mph**, and 5 miles are reachable in 5 minutes (one-step).
    - **Maximum traffic speed observed is 80 mph**, thus κ = 80 miles.
    - c.f. In DCRNN σ = standard deviation of distances between traffic sensors, In STGCN σ = √10, which are ambiguous justification.
  - In our research, traffic sensors are installed on one-way freeways (directed), thus

$$\mathrm{dist}(v_i, v_j) \neq \mathrm{dist}(v_j, v_i).$$

Graph Construction: Final Step

- Construct **co-occurrence similarity matrix** (normalized mutual information) and **distance-based Gaussian proximity matrix**, and conduct **element-wise matrix multiplication**.

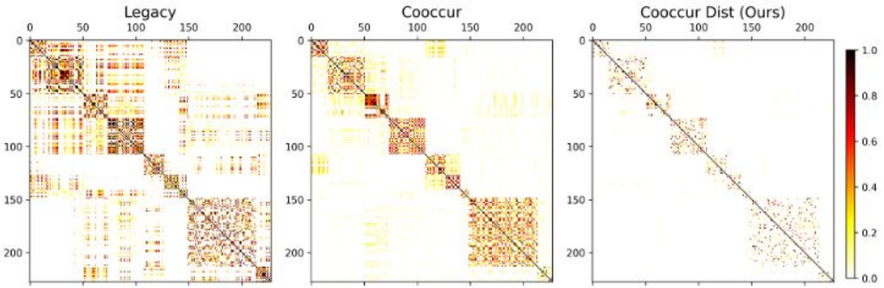


Fig. Adjacency matrix comparison (legacy, co-occurrence, co-occurrence with distance-based Gaussian proximity)

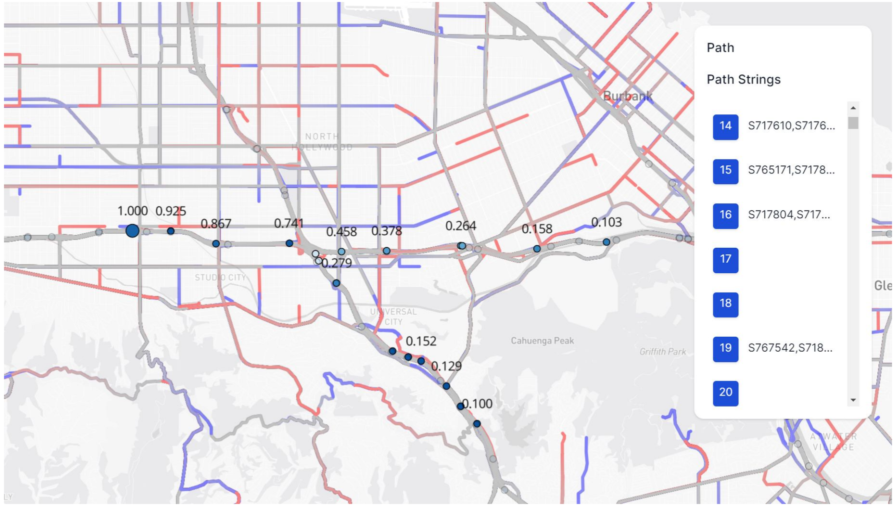


Fig. Sample adjacency matrix values of a target sensor.

(2) Sensor Embedding

- Traffic flow theory (Bruce 1961)– Complex correlation between traffic speed and volume.
  - We assume that such complex correlation is captured during input dense layers.
- Leverage **one-hot sensor embedding** to handle sensor heterogeneity to adjust sensor heterogeneity.
  - Use an embedding layer of *D*-dimension.

Fig. Addition of sensor embedding (SE) to input of encoder and decoder.

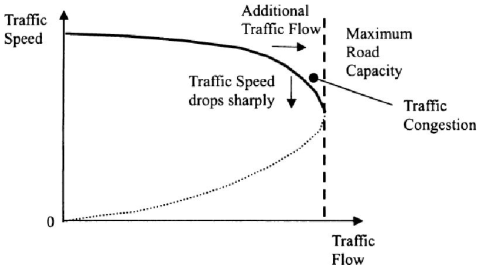


Fig. Traffic flow theory

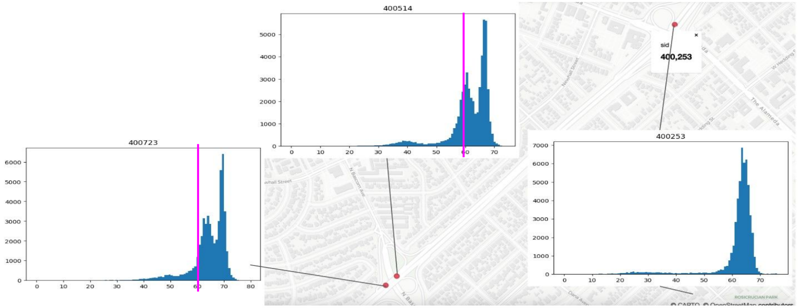


Fig. (Recap) Speed histogram of each traffic sensors. Same 60mph can represent different meanings for each traffic sensors.

(3) Human Activity Embedding

- Human activity pattern from National Household Travel Survey.
  - Our model leverages the frequencies of different human activities from human flow as continuous values.
  - Normalize human activities by maximum value, and apply two-stacked dense layers to generate *D* dimensional embedding.

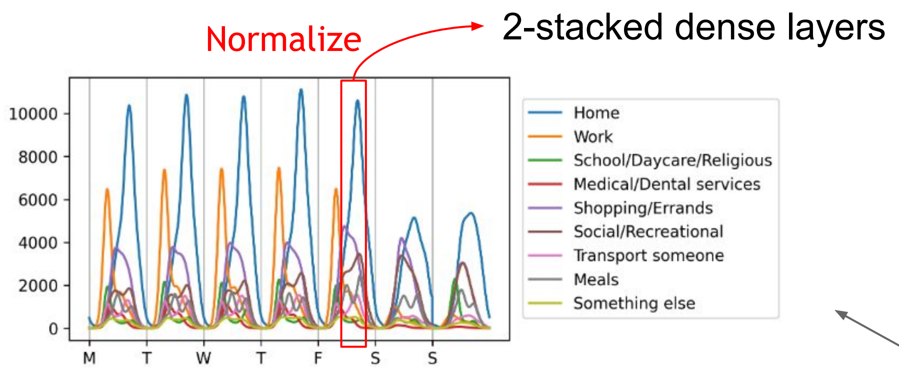


Figure 5: Urban human activity frequencies from the National Household Travel Survey for Activity Embedding.

Fig. Addition of activity embedding (AE) to input of encoder and decoder.

	HOUSEID	PERSONID	TDTRPNUM	STRTTIME	ENDTIME	TRVLCMIN	TRPMILES	TRPTRANS	TRPACCMP	TRPHHACC	...	OBHTNRNT	OBPPOPDN	O
0	30000007	1	1	1000	1015	15	5.244	3	0	0	...	20	750	
1	30000007	1	2	1510	1530	20	5.149	3	0	0	...	30	300	
2	30000007	2	1	700	900	120	84.004	6	0	0	...	40	1500	
3	30000007	2	2	1800	2030	150	81.628	6	0	0	...	20	750	
4	30000007	3	1	845	900	15	2.250	3	0	0	...	20	750	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
923562	40794301	3	2	1658	1705	7	0.781	5	0	0	...	50	3000	
923563	40794301	3	3	1758	1806	8	0.781	5	0	0	...	50	3000	
923564	40794301	3	4	1809	1838	29	9.562	5	1	1	...	50	3000	
923565	40794301	6	1	1631	1657	26	9.115	5	1	1	...	20	300	
923566	40794301	6	2	1809	1838	29	9.562	5	1	1	...	50	3000	

828438 rows × 115 columns

Fig. National Household Travel Survey data (trip distance, purpose, start-end time, etc.), filtered by vehicle use.

Dual-walk Graph Convolution

- Traffic wave theory (Carlos 1994) – traffic congestion can occur **both forward and backward** direction.
- We propose to apply dual-walk graph convolution.
  - c.f. DCRNN propose multi-step dual-walk graph convolution, but insufficient reasoning.

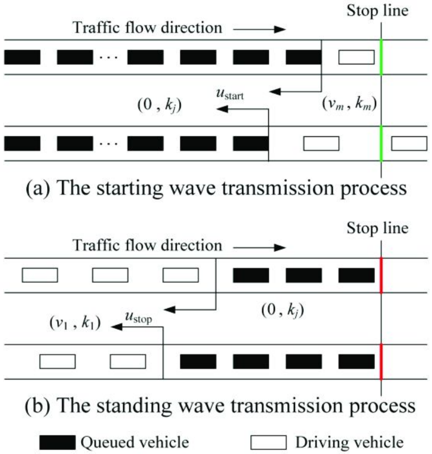


Fig. Traffic wave theory (Carlos 1994)

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

Fig. One-step Dual-walk (forward, backward) graph convolution (ours).

Diffusion Convolution The resulted diffusion convolution operation over a graph signal $\boldsymbol{X} \in \mathbb{R}^{N \times P}$ and a filter $f_{\boldsymbol{\theta}}$ is defined as:

$$\boldsymbol{X}_{:,p} \star_{\mathcal{G}} f_{\boldsymbol{\theta}} = \sum_{k=0}^{K-1} \left(\theta_{k,1} \left(\boldsymbol{D}_O^{-1} \boldsymbol{W} \right)^k + \theta_{k,2} \left(\boldsymbol{D}_I^{-1} \boldsymbol{W}^\intercal \right)^k \right) \boldsymbol{X}_{:,p} \quad \text{for } p \in \{1, \cdots, P\} \quad (2)$$

Fig. c.f. DCRNN demands at least K>=3 steps of multi-step dual-walk graph convolution, which is unnecessary shown in our ablation test.

Graph Convolution with RNN and Transformer

- Final proposed model (UAGCRN, UAGCTransformer):
  - We denote SE + AE as **UA**.
  - Minimal modification to the original GCRNN and GCTransformer.

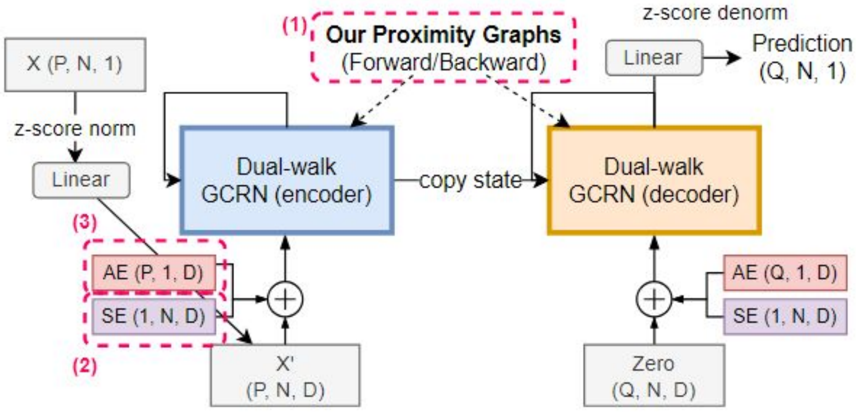


Figure 3: Model Architecture (UA-GCRN)

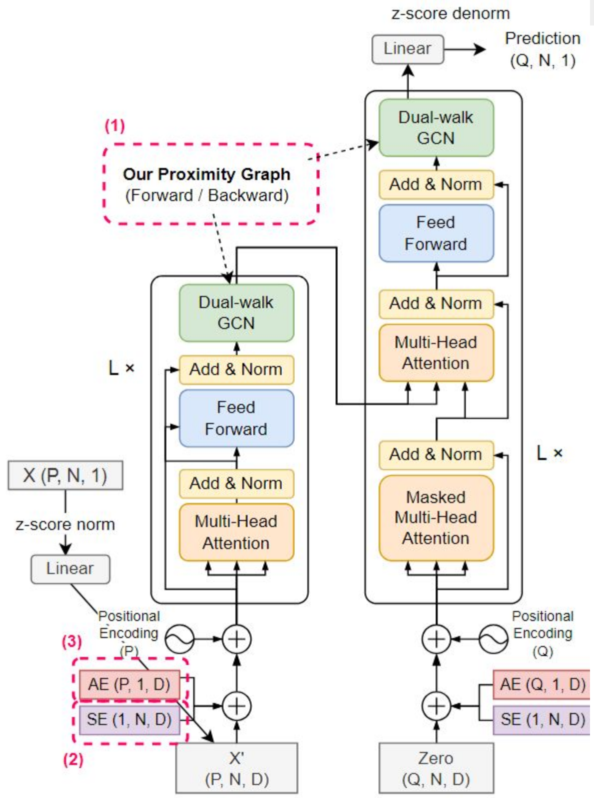


Figure 4: Model Architecture (UA-GCTransformer)

Table of Contents

Data Description and Preprocessing

- Traffic Dataset (which contains accurate sensor location).
- Open Street Map (OSM) Dataset.
  - We adjust accurate sensor location using matching freeway ID on Caltrans Performance Measurement System (PeMS).
- National Household Travel Survey (2017).
- Generate travel paths with different freeway costs (1.0, 0.9, 0.8).

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

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
$N_H^{(Grid)} \times N_W^{(Grid)}$	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}

Evaluation settings

- Evaluation metrics:
  - 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.
- Parameter settings:
  - Input sequence P=12 (1 hour).
  - Output sequence Q=12 (1 hour in METR-LA, PEMS-BAY), Q=9 (45 mins in PEMSD7).
  - Batch size: 32, Embedding size D: 64.
  - Adam optimizer with an initial learning rate of 0.01.
    - Patience of 5 for early stopping, reduce 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. (similar to GMAN default settings)

Performance Comparison

- Performance Comparison
  - † : leveraging our graph
  - ∗ : self-trains the sensor adjacency.
  - **Best** and second best results are represented as **BOLD** and underline.
- UA approach (SE + AE) also improves basic LSTM and TF.
- LSTM, TF performs similar.
- DCRNN† shows improvement over original DCRNN and GTS (same architecture).
- Trainable graph models (GTS, GWNet, GMAN, STEP) does not outperform UAGCRN and UAGCTF except STEP model.
  - STEP takes very long input patch (e.g. P=228×7), which is unfair comparison.

Table 2: Forecasting error comparison.

			Temporal Only					Spatiotemporal								
		Metric	LastRepeat	LSTM	TF	UA-LSTM	UA-TF	DCRNN	DCRNN†	GTS*[29]	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%**

Computational Cost

- Computational cost: UAGCRN < DCRNN < GMAN < UAGCTF
  - Under same learning framework (TensorFlow2) and GPU (RTX3090), batch size, and early stopping condition.
  - UAGCRN consumes less number of diffusion steps than DCRNN (simplified).
  - RNN demands less computational cost than Transformers.

Table 3: Computational cost of METR-LA under the same environment. The number of stacks is $L = 5$ 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

Ablation Study 1 – Our Graph, SE, AE

- Each module contributes performance improvement significantly.
    - Our Graph Construction
    - SE (sensor embedding)
    - AE (activity embedding)

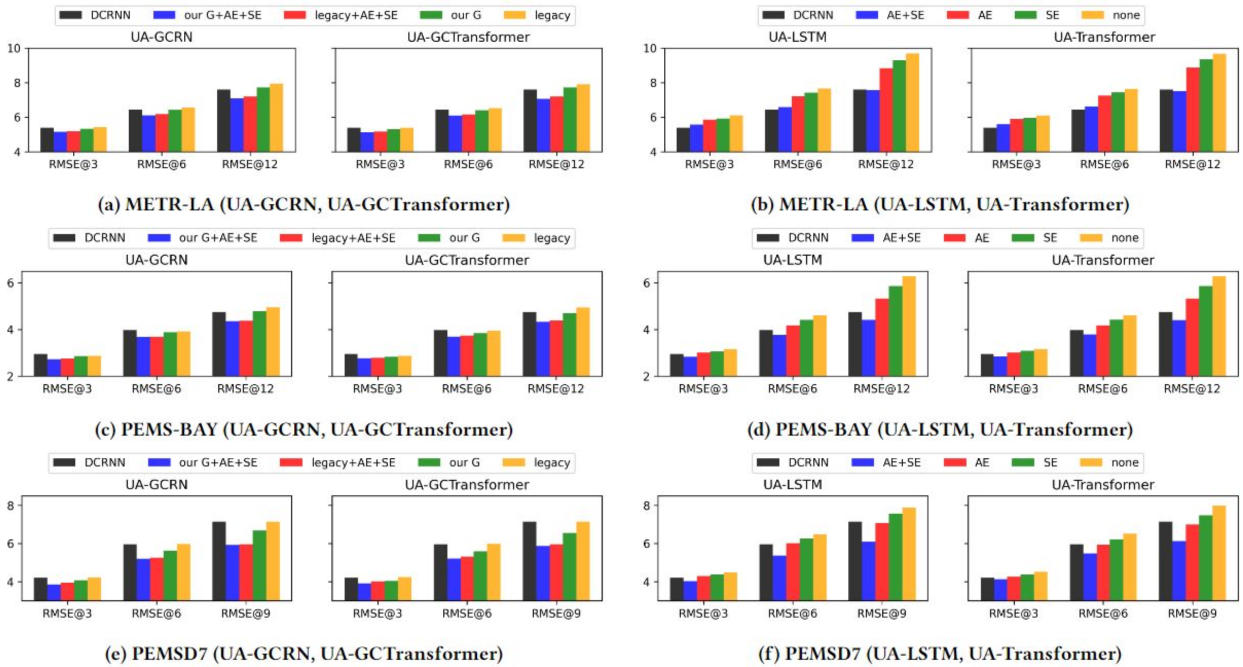


Figure 8: Ablation Test (RMSE) of our models – Our Graph(G), SE, AE.

Ablation Study 2 – # diffusions (K) in DCRNN w/ our graph

- Multiple diffusion steps of DCRNN do not improve performance with our graph.
  - With our graph, additional diffusion steps becomes unnecessary.
  - Vehicle travel pattern follows orient-destination least time-consuming path, not a random walk.
  - c.f. DCRNN construct model on random walk based multi graph convolution.
  - c.f. GMAN leverages Node2vec which leverages random walk path generation.

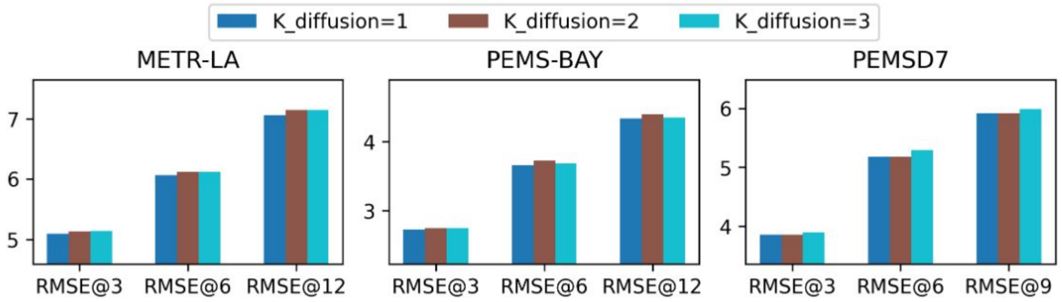


Figure 9: Performance degradation in UADCGRU† as the number of diffusion steps (K) increases.

Ablation Study 3 – TE vs. AE

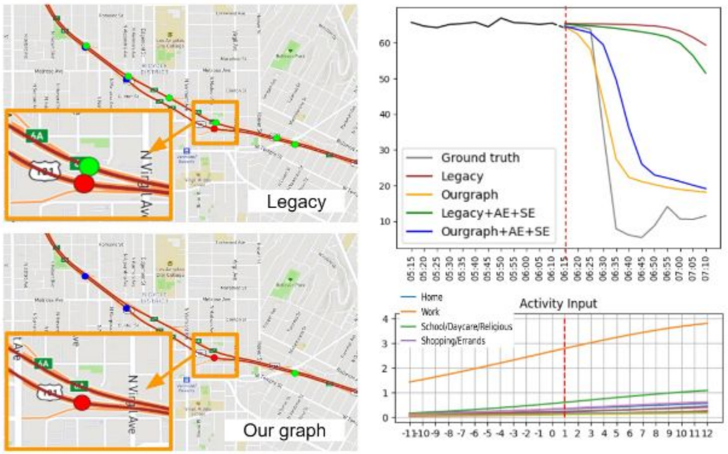
- Comparison between TE vs. AE
  - Conventional timestamp embedding (TE) [DCRNN, GMAN, PDFormer]
    - Weekday one-hot (7):
      - E.g. Mon ⇒ [1, 0, 0, …], Tue ⇒ [0, 1, 0, …]
    - Time-of-a-day one-hot (12×24):
      - E.g. 0:00 ⇒ [1, 0, 0, …], 0:05 ⇒ [0, 1, 0, …]
    - Concatenate and apply 2-stacked dense layers.
  - Activity embedding (AE — our proposed).
- TE still performs better than AE.
  - National Survey based AE may not reflect accurate human activity for each region (Los Angeles, Santa Clara)
  - However, AE reflect **continues nature of time.**
  - AE approach has more potential to be generalize the activity-traffic correlation which is limited with TE.
    - National holidays, seasonal vacations, etc.

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.

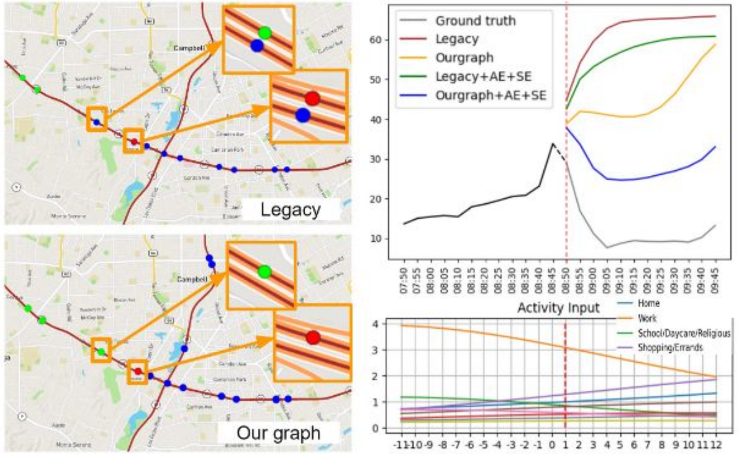
		STEP	UAGCRN†		UAGCTF†	
			TE+SE	AE+SE	TE+SE	AE+SE
METR-LA	MAE3	**2.61**	2.62	2.64	2.63	2.63
	RMSE3	**4.98**	5.00	5.09	5.09	5.07
	MAE6	2.96	**2.94**	2.97	2.95	2.96
	RMSE6	5.97	**5.97**	6.08	6.05	6.04
	MAE12	3.37	**3.31**	3.35	3.35	3.34
	RMSE12	**6.99**	7.02	7.12	7.10	7.02
PEMS-BAY	MAE3	**1.26**	1.28	1.30	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
	MAE12	**1.79**	1.88	1.87	1.86	1.86
	RMSE12	**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
	MAE9	2.95	**2.88**	2.92	2.89	2.90
	RMSE9	6.03	**5.88**	5.90	5.88	5.91

Case Study

- Case study: UAGCRN on METR-LA and PEMS-BAY.
- 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.



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



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

Application – Activity-based traffic congestion estimation

- Our model finds correlation between **activity frequency** and **traffic congestion.**
- We can observe which sensors are more affected by specific urban human activity.

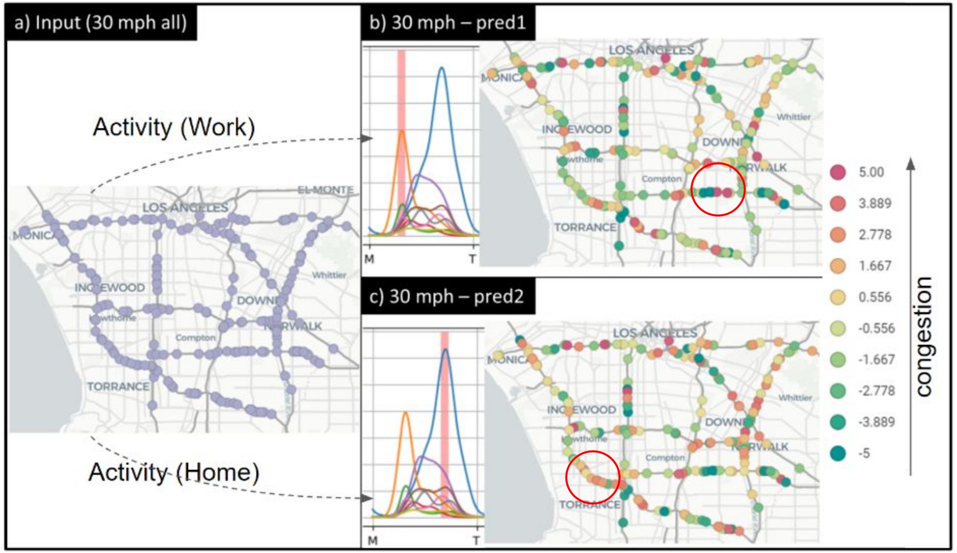


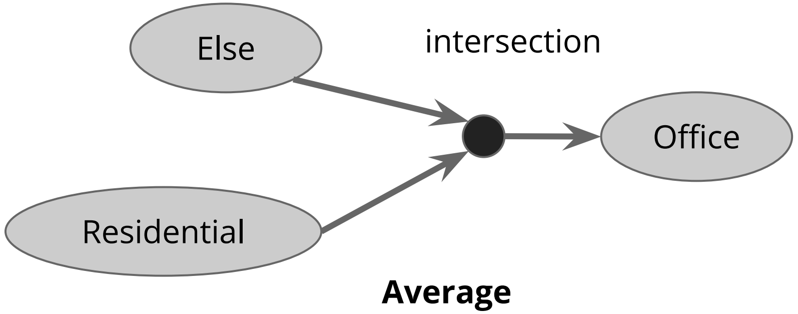
Figure 11: Sensor Reactions Based on Activity Information with UAGCRN (Red/Green: more/less congestion)

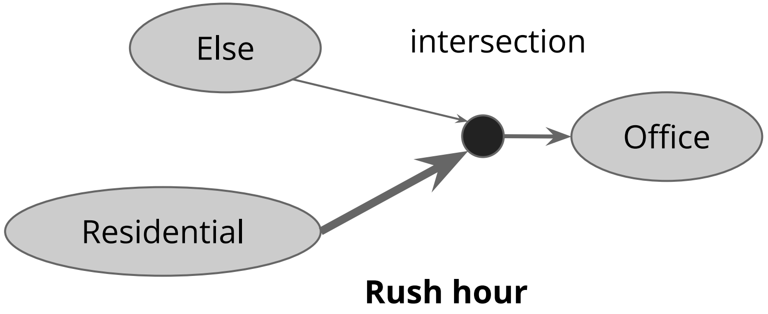
Table of Contents

Discussion

- Dynamic Graph Construction
  - Realistic urban simulation: Land use, Building types, POIs.
  - Incorporate orient-destination travel demand prediction model for path generation.
  - Incorporate traffic simulators (e.g. SUMO).

Fig. Dynamic graph (varies over time).





Discussion

- Simulated travel survey
  - Large Language Model(LLM) based simulated agents behavior in a town [JS Park]
    - "Generative agents: Interactive simulacra of human behavior." UIST (2023) Cited by 152.

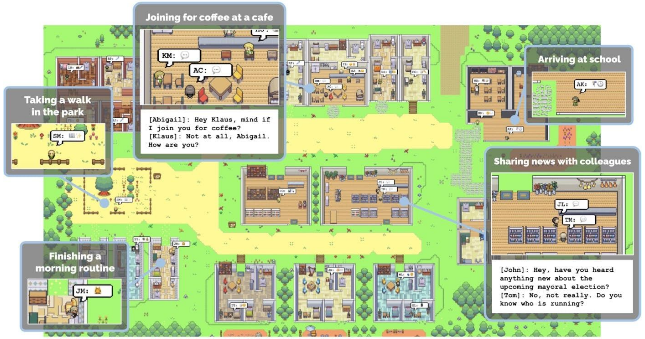
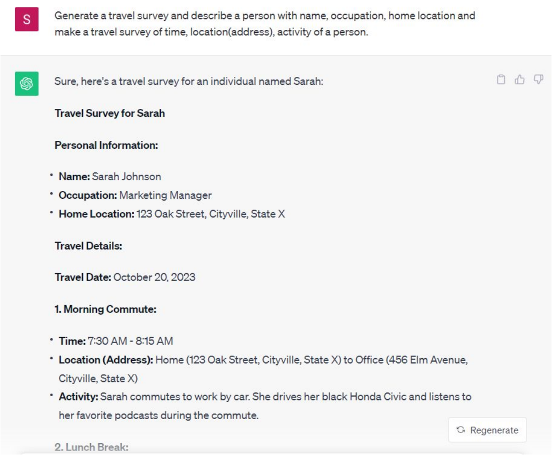


Fig. ChatGPT generated agents interact in a game world [JS Park].



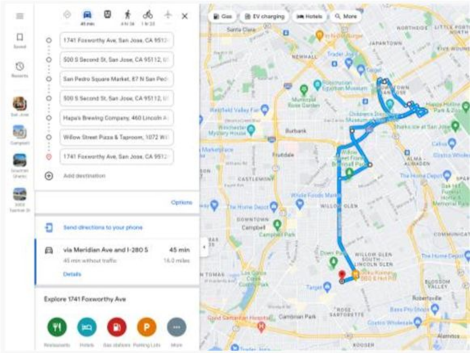


Fig. ChatGPT-generated probable travel survey in Santa Clara.

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.
  - Travel path generation with A* algorithm, co-occurrence and distance-Gaussian sensor adjacency matrix.
  - Sensor-specific one-hot encoding.
  - Human activity embedding.
- Minimal modifications to 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.

Q & A