Continuous-Time and Multi-Level Graph Representation Learning for Origin-Destination Demand Prediction

Liangzhe Han[1], Xiaojian Ma[1], Leilei Sun[1], Bowen Du[1], Yanjie Fu[2], Weifeng Lv[1], Hui Xiong [3]

Introduction

Origin-Destination (OD) demand prediction aims to predict the demand between two traffic nodes.Although there are some attempts on OD demand prediction, two important issues have rarely been discussed.

- **First, historical transactions are generally aggregated into demand snapshots, which will result in inevitable information loss.**
- **Second, the spatial dependency in prior studies is always manually designed, which is intuitive but incomplete.**

To address the above issues, this study proposes a novel Continuous-time and Multi-level dynamic graph representation learning framework for Origin-Destination demand prediction (CMOD).

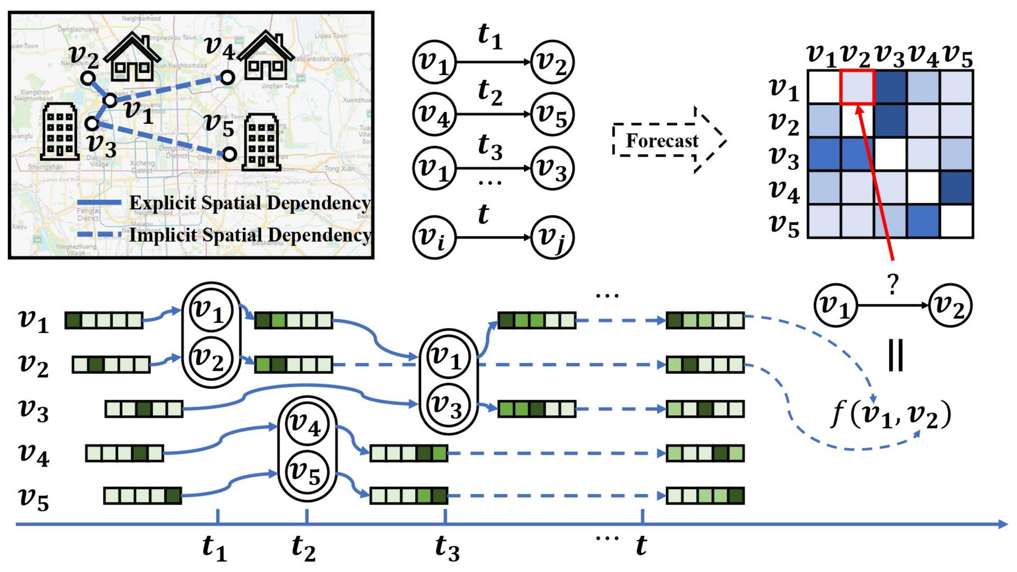

Figure 1: OD demand prediction with continuously evolving node representations.

Formulation

- **DYNAMIC TRANSACTION GRAPH. Transcations contain origin, destination and departure time, which are organized as a continuous-time dynamic graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is a finite set of N traffic nodes; $\mathbb{E}$ is the set of M timestamped transactions. An edge $e_m = (v_m^o, v_m^d, t_m)$ represents a passenger from v_m^o to v_m^d at time t_m.**
- **OD DEMAND MATRIX. The OD demand matrix between t and $t+\tau$ is denoted as $\mathbf{Y}^{t:t+\tau} \in \mathbb{R}^{N\times N}$.The (i,j)-entry of $\mathbf{Y}^{t:t+\tau}$ represents how many passengers travel from v_i to v_j between t and $t+\tau$: $\mathbf{Y}_{i,j}^{t:t+\tau} = |\{e_k | v_k^o = v_i \wedge v_k^d = v_j \wedge t \le t_k < t+\tau\}|$, where $|\cdot|$ is the size of a set.**
- **OD DEMAND PREDICTION PROBLEM. Given historical transaction records, OD demand matrix in the next period of time is calculated as follow:**

$$\widehat{\mathbf{Y}}^{t:t+\tau} = f(\mathcal{G}_t, \mathbf{F}, \mathbb{W}),$$

where $\mathbb{W}$ is the set of learnable parameters.

Dataset

We evaluated our methods on two real world datasets, BJSubway and NYTaxi. Detailed statistic information of these datasets is shown in table 1.

Table 1: Statistic information of datasets

Dataset	BJSubway	NYTaxi
#Nodes	268	63
#Orders	279,227,618	38,498,427
#Train Days	42	139
#Validation Days	7	21
#Test Days	7	21
Average Demand	2.1694	1.1164
Zero Order Ratio	54.84%	66.15%

Methodology

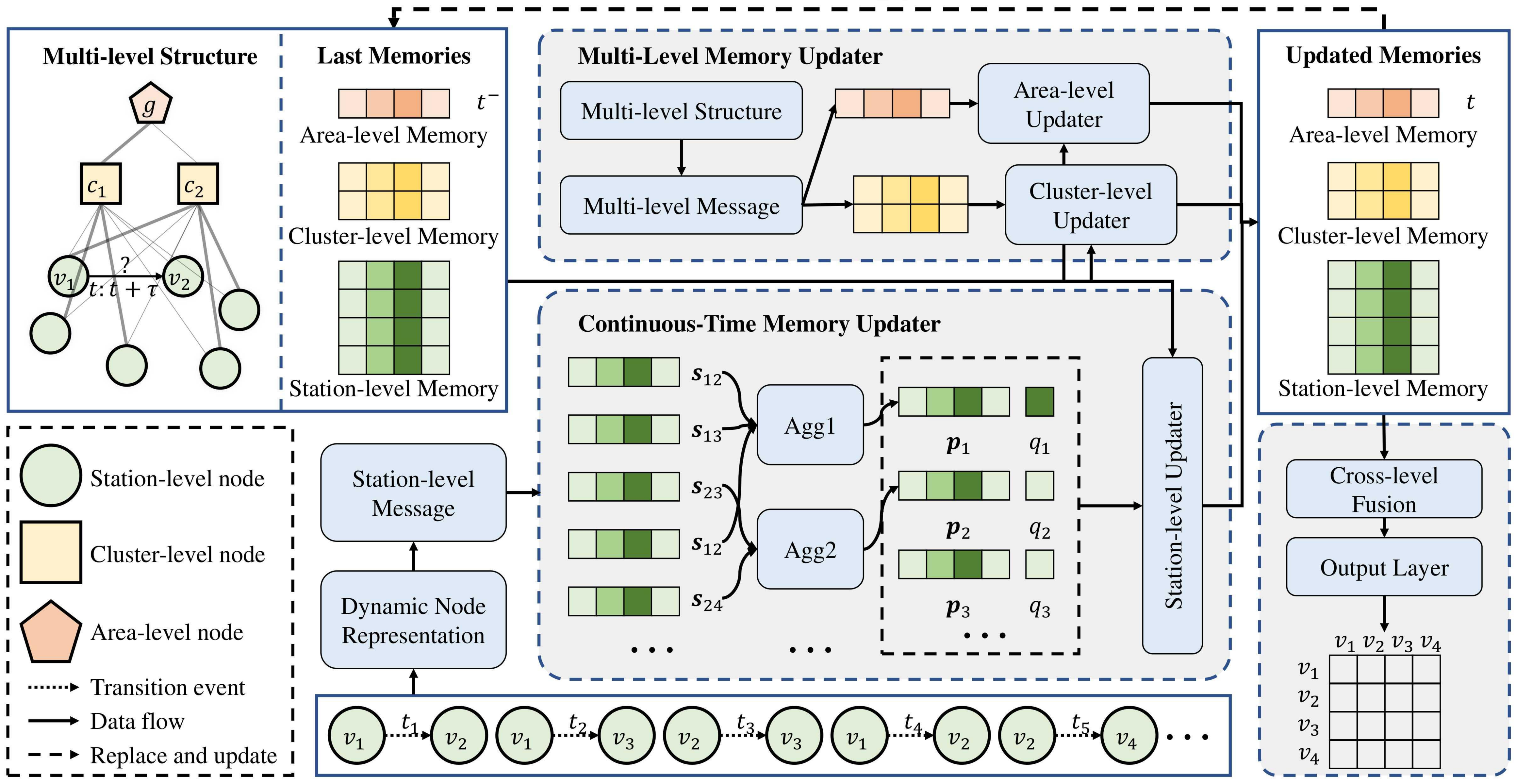

Figure 2: The overall framework of CMOD. The core idea of the framework is to maintain multi-level memories for nodes. The multi-level structure is designed to capture spatial dependency between traffic nodes. When transitions happen, this framework will update these memories with these streaming events. And the updated memories, which compress all historical transactions and represent real-time node status, are utilized for the final prediction.

- **Continuous-Time Node Representation. CMOD is directly built on raw transition records, which views time information as continuous features and maintains continuous-time evolving representation(memory) for each traffic node .**
- **Multi-level Structure. As shown in top-left of Figure 2, station-level nodes are aggregated to virtual cluster-level nodes and cluster-level nodes are aggregated to the virtual area-level node. Their memory are update as figure 3 shows.**

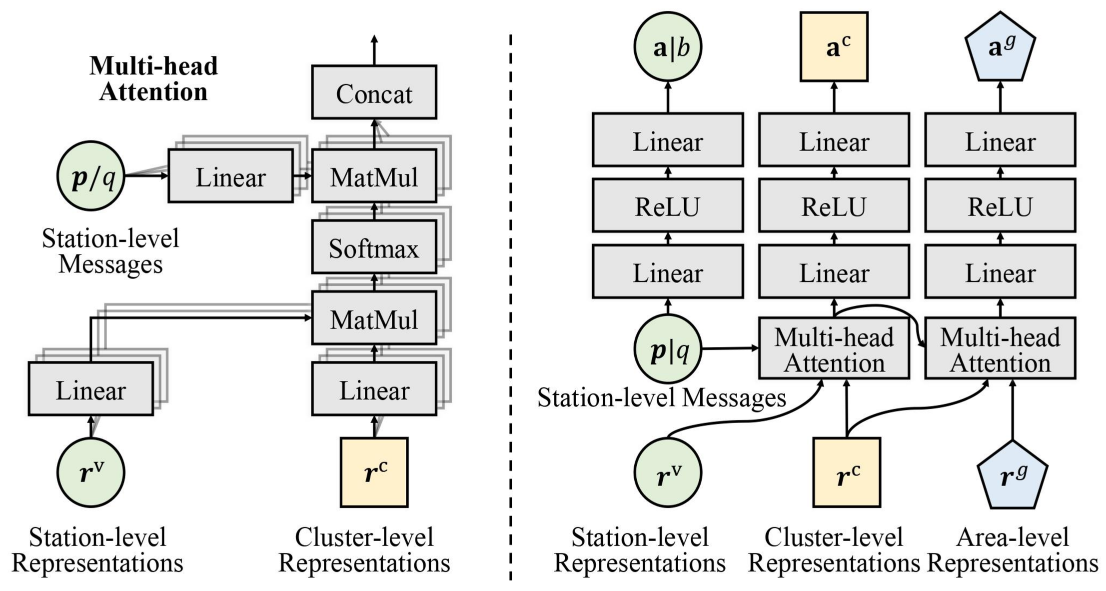

Figure 3:Multi-level memory updater

Experiment

Table 2: Comparison with baselines.

Dataset	Method	All OD Pairs MAE ↓	All OD Pairs RMSE ↓	All OD Pairs PCC ↑	OD Pairs with Demand above Average MAE ↓	OD Pairs with Demand above Average RMSE ↓	OD Pairs with Demand above Average PCC ↑
BJSubway	HA	2.9003	8.1266	0	8.0378	18.3277	0
	LR	1.9396	5.3547	0.7521	6.0566	11.7181	0.7322
	XGBoost	1.8048	5.7709	0.7040	5.9098	12.9627	0.6449
	GEML	1.7291±0.0123	4.6018±0.1138	0.8279±0.0075	5.3002±0.0982	10.1491±0.2983	0.8083±0.0086
	DNEAT	1.4706±0.0099	5.7384±0.0311	0.7237±0.0033	5.4476±0.0365	13.0661±0.0798	0.6488±0.0055
	TGN	2.1031±0.1629	5.8927±0.5148	0.6755±0.0659	6.5592±0.3331	13.0607±1.1772	0.6455±0.0781
	DyRep	-	-	-	-	-	-
	CMOD	**1.4475±0.0202**	**3.6890±0.0319**	**0.8911±0.0020**	**4.5068±0.0437**	**8.1441±0.0697**	**0.8773±0.0021**
NYTaxi	HA	1.4593	2.6569	0	3.6041	5.7289	0
	LR	0.6907	1.3611	0.8586	1.9939	2.8069	0.8164
	XGBoost	0.6881	1.3555	0.8599	1.9895	2.8052	0.8185
	GEML	0.6476±0.0033	1.3432±0.0093	0.8662±0.0015	1.8867±0.0138	2.7587±0.0198	0.8201±0.0025
	DNEAT	0.6495±0.0025	1.5179±0.0172	0.8252±0.0040	2.1922±0.0292	3.2834±0.0685	0.7581±0.0104
	TGN	0.6516±0.0142	1.2947±0.0330	0.8747±0.0057	1.8387±0.0235	2.6435±0.0503	0.8311±0.0094
	DyRep	0.6094±0.0032	1.2164±0.0089	0.8892±0.0013	1.7844±0.0296	2.5074±0.0398	0.8528±0.0017
	CMOD	**0.5926±0.0026**	**1.1795±0.0023**	**0.8959±0.0004**	**1.7244±0.0091**	**2.4263±0.0089**	**0.8618±0.0006**

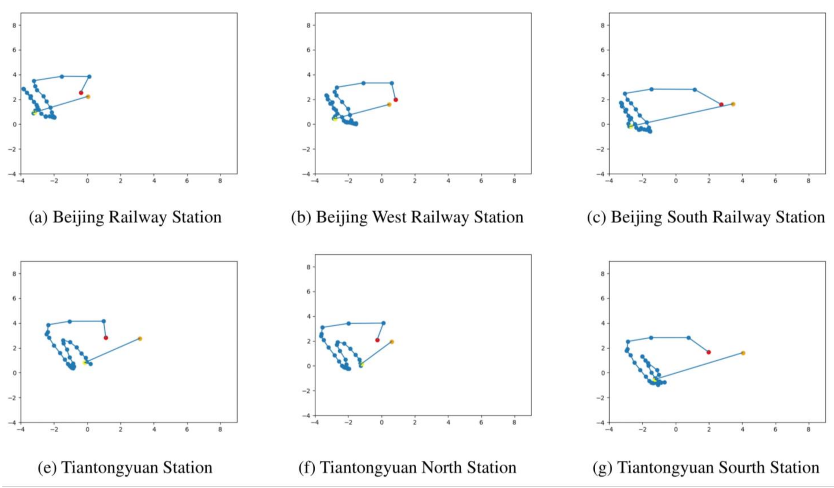

Figure 4:Intrepretation of evolving dynamic station representations

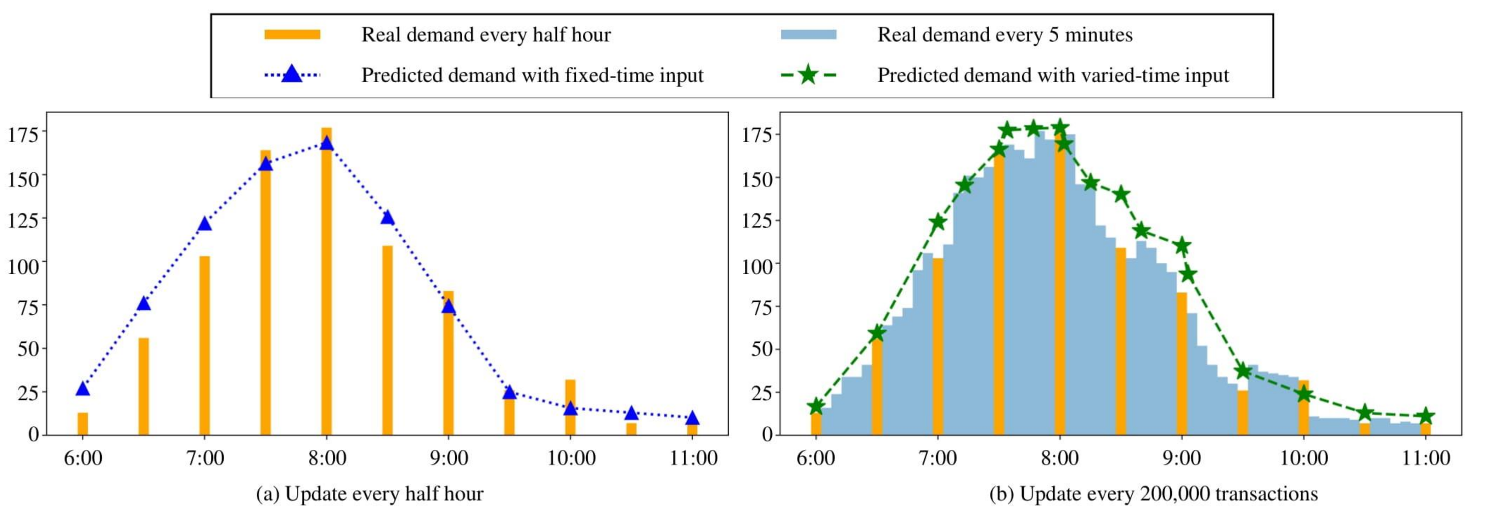

Figure 5:Prediction in two different settings

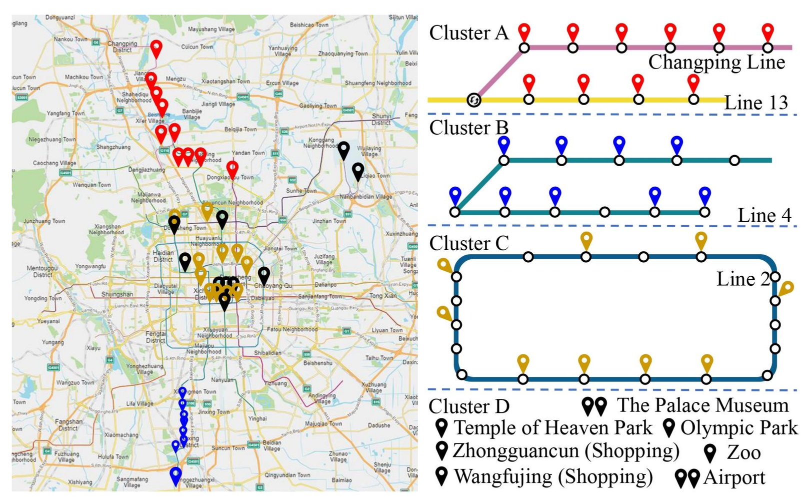

Figure 6:Intrepretation of discovered clusters

- **Table 2 summarizes the performance of all baselines, including traditional machine learning methods, snapshot based OD prediction methods and dynamic graph methods.**
- **Figure 4 shows some representations of stations in BJSubway, demonstrates that CMOD can automatically discover similar patterns from adjacent stations without predefined geographical information.**
- **Figure 5 shows that the time granularity of CMOD's input need not to set explicitly. .Thus, CMOD is able to update memories with varied timespans and predict demand whenever the memories are updated.**
- **The highest weighted station-level nodes are selected to illustrate their locations in figure 6, demonstrate that the multi-level structure can adaptively aggregate station-level nodes to clusters, establish relations among traffic nodes and benefit the final prediction**