



Road Network Representation Learning with Vehicle Trajectories

Stefan Schestakov¹(✉), Paul Heinemeyer¹, and Elena Demidova^{2,3}

¹ L3S Research Center, Leibniz Universität Hannover, Hannover, Germany
{schestakov,pheinemeyer}@L3S.de

² Data Science and Intelligent Systems Group (DSIS), University of Bonn,
Bonn, Germany
elena.demidova@cs.uni-bonn.de

³ Lamarr Institute for Machine Learning and Artificial Intelligence,
Bonn, Germany
<https://lamarr-institute.org/>

Abstract. Spatio-temporal traffic patterns reflecting the mobility behavior of road users are essential for learning effective general-purpose road representations. Such patterns are largely neglected in state-of-the-art road representation learning, mainly focusing on modeling road topology and static road features. Incorporating traffic patterns into road network representation learning is particularly challenging due to the complex relationship between road network structure and mobility behavior of road users. In this paper, we present TrajRNE – a novel trajectory-based road embedding model incorporating vehicle trajectory information into road network representation learning. Our experiments on two real-world datasets demonstrate that TrajRNE outperforms state-of-the-art road representation learning baselines on various downstream tasks.

1 Introduction

Effective general-purpose representations of road networks are essential for critical machine learning applications in mobility and smart cities, such as traffic inference, travel time estimation, and destination prediction. This demand has recently inspired numerous research works on road network representation learning (e.g., [1, 16, 17, 19]). Whereas existing approaches primarily utilize road network topology and static road features, they often fail to capture complex traffic patterns and mobility behavior of road users. A rich source of complex spatio-temporal traffic patterns, traffic flows, actual-driven speed, and driver road preferences are vehicle trajectories. Thus, integrating vehicle trajectory information into the road network representation can provide valuable information for mobility and smart city applications.

Previous road representation learning approaches (e.g., [1, 16, 17, 19]) have two substantial shortcomings. First, state-of-the-art methods utilize conventional graph representations (e.g., [9, 13, 15]), which do not consider complex road relationships. For example, for a road leading to an intersection, the importance of the following roads is not equal and depends on user mobility behavior. Second, state-of-the-art road representation models learn static road features, e.g., road

type and speed limit, to infer traffic patterns. However, these features do not directly reflect dynamic traffic conditions. For example, roads with the same speed limit can have vastly different traffic patterns depending on the traffic volume. Recently, few approaches attempted to incorporate trajectories into road network representation learning. Wang et al. [16, 17] supplemented random walks with real-world trajectories for learning geo-locality. Wu et al. [19] utilized trajectory data as a supervision signal for graph reconstruction. Further, Chen et al. [1] refined previously learned road embeddings with a route recovery and trajectory discrimination supervision objective using a transformer model. However, existing approaches do not explicitly incorporate trajectory data into their model design and thus fail to incorporate complex traffic and mobility patterns.

We observe two substantial challenges for general-purpose road representation learning. First, conventional graph representation learning methods [9, 13, 15] are inadequate for road network modeling, as they assume network homophily and do not consider heterogeneous properties of connected roads and complex road relationships. In contrast, connected roads, e.g., a secondary road connected to a primary road, can exhibit highly diverse traffic patterns. Thus, the first challenge is to adapt graph representation methods to road networks with heterogeneous traffic patterns on connected roads. Second, a challenge is to systematically incorporate vehicle trajectories into road representation learning to extract and represent dynamic traffic patterns and complex mobility behavior.

In this paper, we propose a novel Trajectory-based Road Network Embedding model (TrajRNE). TrajRNE includes two modules. First, we propose a novel Spatial Flow Convolution (SFC). SFC aggregates road feature representations based on transition probabilities extracted from vehicle trajectories. Thus, SFC automatically differentiates between relevant and irrelevant road network nodes indicated by the mobility behavior. Moreover, we increase the SFC receptive field by considering the traffic flow of k -hop neighbors. This approach facilitates aggregation of relevant neighbors located at a longer distance without over smoothing with non-relevant neighbors. Second, we propose a novel Structural Road Encoder (SRE) leveraging multitask learning to capture topology, structure, and dynamic traffic. Whereas state-of-the-art road embeddings learn topology using random walks or shortest paths, they do not effectively capture mobility behavior. In contrast, TrajRNE adopts random walks based on the transition probability extracted from real-world trajectories to capture geo-locality and mobility patterns.

In summary, the contributions of our work are as follows:

- We introduce Spatial Flow Convolution and Structural Road Encoder to capture traffic characteristics of road networks from vehicle trajectories.
- We propose TrajRNE¹ – a novel road network representation learning approach, effectively capturing traffic patterns with SFC and SRE methods.
- Our evaluation demonstrates that TrajRNE enables effective general-purpose road network representations. TrajRNE consistently outperforms state-of-the-art baselines on four downstream tasks and two real-world datasets.

¹ Code available at: <https://github.com/sonout/TrajRNE>.

2 Problem Definition

In this section, we first present the notations and then formally define our task.

Definition 1. (Road Network). We define a road network as a directed graph $G = (\mathcal{V}, \mathcal{A}, \mathcal{F})$. \mathcal{V} is a set of nodes, where each node $v_i \in \mathcal{V}$ represents a road segment. \mathcal{A} is the adjacency matrix, where $\mathcal{A}_{ij} = 1$ implies that a road segment v_j directly follows a road segment v_i , and $\mathcal{A}_{ij} = 0$ otherwise. A road network has a feature set $\mathcal{F} \in \mathbb{R}^{|\mathcal{V}| \times f}$ representing road segment features with dimension f .

Definition 2. (Trajectory). A trajectory T is a sequence of points representing geographic coordinates from the route driven by a vehicle: $T = [p_1, p_2, \dots, p_{|T|}]$, where $p_i = (lon_i, lat_i)$ is the i -th point with the longitude lon_i and latitude lat_i and $|T|$ is the trajectory length.

Given a road network G , we can map a trajectory T to the road network using a map matching algorithm [20], thus obtaining a sequence of road segments.

Definition 3. (Road Segment Sequence). A road segment sequence $R = [v_1, v_2, \dots, v_N]$ represents the underlying route of a trajectory on a road network, where each $v_i \in \mathcal{V}$ denotes a road segment in the road network $G = (\mathcal{V}, \mathcal{A}, \mathcal{F})$.

In this work, we target the problem of learning a general-purpose representation of road networks beneficial for various downstream tasks.

Definition 4. (Road Network Representation Learning). Given a road network $G = (\mathcal{V}, \mathcal{A}, \mathcal{F})$ and a set of trajectories $\mathcal{T} = \{T_i\}_{i=1,2,\dots,|\mathcal{T}|}$, our objective is to learn a representation r_i for each road segment through an unsupervised model F . As a result, we obtain the set of all road representations $S = F(G, \mathcal{T}) \in \mathbb{R}^{|\mathcal{V}| \times d}$ with dimension d .

3 TrajRNE Approach

In this section, we introduce our proposed Trajectory Road Network Embedding Model (TrajRNE) to learn effective, general-purpose embeddings of road segments in an unsupervised manner. As illustrated in Fig. 1, TrajRNE incorporates two modules, the Spatial Flow Convolution (SFC) and the Structural Road Encoder (SRE). In the following, we present these modules in more detail.

3.1 Spatial Flow Convolution

The Spatial Flow Convolution aggregates roads based on the flow probabilities provided by trajectories. Moreover, we designed the SFC to aggregate over a k -hop neighborhood to leverage distant dependencies. Standard Graph Convolutional Networks (GCNs) commonly assume network homophily, i.e., connected nodes are more similar than distant nodes. However, road networks possess complex dependencies between roads. On the one hand, consecutive roads can indicate different traffic patterns. On the other hand, traffic patterns on distant road

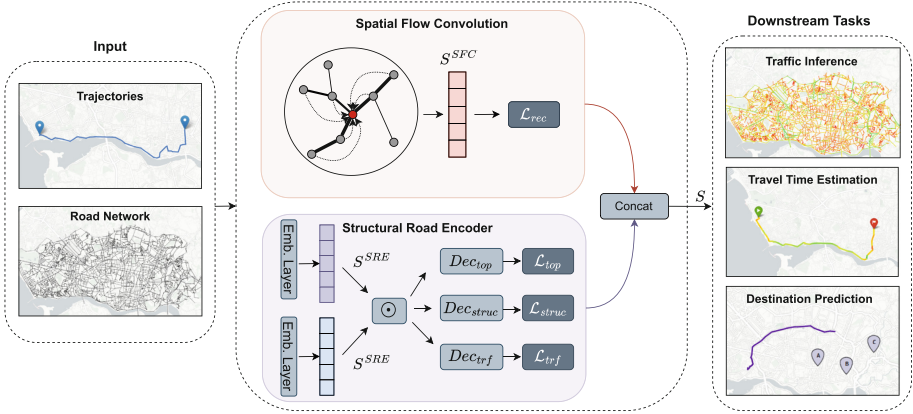


Fig. 1. The proposed TrajRNE architecture incorporating two modules: Spatial Flow Convolution and Structural Road Encoder.

segments can be correlated. Therefore, GCNs are not suitable for learning road network representations.

Inspired by Li et al. [10], who utilized trajectory flows to aggregate spatial traffic information for flow prediction, we design the Spatial Flow Convolution. In contrast to [10], we increase the receptive field by considering the traffic flow of k-hop neighbors and aggregating them within a single layer. This enables us to design an aggregation function, which selectively aggregates local and distant roads based on their importance provided by trajectory flows. We depict our Spatial Flow Convolution in Fig. 2 and compare it to a two-layer GCN. The GCN (left) aggregates all neighbors equally and needs to be stacked, which can lead to over smoothing [12]. The Spatial Flow Convolution (right) aggregates roads based on the vehicle flows (indicated by the thickness of the edges) and thus can weight the aggregation of roads based on their importance. Thus, we consider even distant relationships and tackle the issue of over smoothing by considering only important roads for aggregation.

To obtain the vehicle flow between the roads, we introduce the road transition probability p . Given two road segments v_i and v_j , the road transition probability is the probability of visiting v_j when v_i has been visited. We formally define the road transition probability by:

$$p(v_j|v_i) = \frac{p(v_i \cap v_j)}{p(v_i)}. \tag{1}$$

We estimate p by aggregating the number of transitions in historical trajectories:

$$\hat{p}(v_j|v_i) = \frac{\#transitions(v_i \rightarrow v_j) + \mathcal{A}_{i,j}}{\#total_visits(v_i) + \sum_{k=0}^{|V|} \mathcal{A}_{i,k}}, \tag{2}$$

where \mathcal{A} is the adjacency matrix and $\mathcal{A}_{i,j}$ is 1 when the road segment v_j directly follows v_i . That way we keep road segment connections, even in case of sparsity of



Fig. 2. Left: Graph convolution with two layers, aggregating each node with equal importance. Right: Our proposed approach of aggregating nodes based on traffic flow and hop distance for a maximum hop distance of two.

trajectories. Further, we build a road transition probability matrix $P \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ containing the transition probability of every road segment pair, i.e., $P_{i,j} = \hat{p}(v_j|v_i)$. The superscript k indicates that for the construction of P we consider all road segment pairs (v_i, v_j) within a k -hop distance, e.g., for P^3 we consider the transition probabilities for roads up to three hops away.

We leverage the transition probability matrix P^k to perform graph convolutions over road networks. We define the spatial flow convolution formally as

$$S^{SFC} = \sigma(P^k \mathcal{F} W), \quad (3)$$

where W is a trainable weight matrix, \mathcal{F} is the set of road features, σ an activation function and S^{SFC} are the obtained road representations.

To train this module in an unsupervised way, we employ the graph reconstruction task. Thus, having obtained the road representations S^{SFC} , we try reconstructing the original adjacency matrix \mathcal{A} :

$$\hat{\mathcal{A}} = \text{sigmoid}(S^{SFC} \cdot S^{SFC \top}). \quad (4)$$

We employ mean squared error loss to compute the reconstruction loss:

$$\mathcal{L}_{rec} = \|\mathcal{A} - \hat{\mathcal{A}}\|^2. \quad (5)$$

The advantage of the reconstruction loss is that it forces the road representations S^{SFC} to learn effective road characteristics and the road network topology.

3.2 Structural Road Encoder

The Structural Road Encoder encodes structural and dynamic traffic properties by training on a multitask prediction objective in a contrastive way. More precisely, we predict whether two road segments are similar regarding three characteristics: topology, network structure, and traffic.

- **Topology (*top*):** To learn the topology, we predict whether two road segments co-occur on a random walk. However, random walks do not represent typical road users. Therefore, we propose to weight the random walks based on the transition probabilities provided by vehicle trajectories. More specifically, we utilize the transition probability matrix P^1 for the first-degree neighborhood and use this matrix as the transition probability source for the random walk generation. The resulting trajectory-weighted random walks reflect the geo-locality of the road network and user mobility behavior.
- **Network structure (*struc*):** In this task, we predict if the node degree of two road segments is the same. The node degree is an essential structural road network feature. It helps to distinguish roads with only one consecutive road segment from, e.g., roads followed by complex intersections.
- **Traffic (*trf*):** For the third task, we predict whether two road segments have similar traffic. For the traffic label, we utilize a traffic feature extracted from trajectories, i.e., mean traffic speed or volume. As those features are continuous, we divide them into ten equally sized categories and predict whether two road segments fall into the same category. In contrast to previous works learning static road features, we train on features extracted from trajectories, which reflect real-world traffic patterns.

For the training data generation, we sample n trajectory-weighted random walks per road segment, with a walk length of l and a context window of w . For each pair within a window, we set the topology label Y_{top} to 1 and obtain the structure label Y_{struc} and traffic label Y_{trf} . Further, for each positive sample, we create n_{neg} negative samples by randomly selecting road segment pairs, setting $Y_{top} = 0$ and obtaining Y_{struc} and Y_{trf} .

For the SRE training, we input two one-hot-encoded vectors $v_i, v_j \in \mathbb{R}^{|\mathcal{V}|}$ indicating the index of the road segment and encode the input into dense vectors.

$$S_i^{SRE} = Emb(v_i), \quad (6)$$

where S_i^{SRE} is the dense vector representation of the road segment v_i and Emb is the embedding layer, modeled as a fully connected layer. To predict the task labels, we employ the Hadamard product to aggregate the two road embeddings S_i^{SRE} and S_j^{SRE} and input the resulting vector into a task-specific decoder. Then for each $task \in \{top, struc, trf\}$ we obtain a probability output \mathcal{P} :

$$\mathcal{P}_{task}(v_i, v_j) = Dec_{task}(S_i^{SRE} \odot S_j^{SRE}), \quad (7)$$

where \odot represents the Hadamard product, and Dec is a task-specific decoder, which we model using a fully connected layer and a sigmoid activation function, i.e., $Dec(x) = sigmoid(FC(x))$. Given the task labels, we can formulate the loss functions for each task as the binary cross-entropy loss:

$$\mathcal{L}_{task}(v_i, v_j) = -[Y_{task} \cdot \log(\mathcal{P}_{task}) + (1 - Y_{task}) \cdot \log(1 - \mathcal{P}_{task})]. \quad (8)$$

The overall loss function of the SRE is defined as the weighted sum of \mathcal{L}_{top} , \mathcal{L}_{struc} and \mathcal{L}_{trf} with the corresponding weights $\lambda_{top} + \lambda_{struc} + \lambda_{trf} = 1$:

$$\mathcal{L}_{SRE} = \lambda_{top} \cdot \mathcal{L}_{top} + \lambda_{struc} \cdot \mathcal{L}_{struc} + \lambda_{trf} \cdot \mathcal{L}_{trf}. \quad (9)$$

3.3 TrajRNE Overview

In our proposed TrajRNE model, we train the SFC and SRE modules independently with distinct training objectives. We concatenate the module representations to obtain the final road representation: $S = S^{SFC} \oplus S^{SRE}$, where \oplus is the concatenation operator. As the SFC and SRE representations contain complementary information, they induce more information into the final road representations, making them more effective and general-purpose. Moreover, in contrast to previous work, we incorporate traffic and mobility behavior into the TrajRNE model design, which is essential for various downstream tasks.

4 Experimental Evaluation

The aim of the evaluation is threefold. First, we aim to compare TrajRNE with state-of-the-art unsupervised road embedding models on various road network-related downstream tasks. Second, we aim to evaluate ablation versions of TrajRNE. Third, we aim to assess the impact of the k parameter of SFC, as it influences the receptive field.

4.1 Datasets

We select the trajectory and road network datasets for two cities, namely Porto² and San Francisco³. The road networks are extracted from OpenStreetMap⁴. We preprocess the trajectory data. In particular, we prune trajectories outside the bounding box of the respective city and remove trajectories containing less than 10 points. Further, we map-match the trajectories [20] to obtain the road segment sequences. Table 1 summarizes the dataset statistics.

4.2 Baselines

We employ state-of-the-art road network representation models and graph representation learning approaches as baselines. For road network representation models, we evaluate **RFN** [7], **IRN2Vec** [16], **HRNR** [19] and **Toast** [1] as baselines. For graph representation learning approaches, we select **GCN** [9], and **GAT** [15] as baselines. We employ the graph reconstruction task proposed in [8] to train GCN and GAT in an unsupervised fashion. We use the parameter values given in the original papers. Note that as we aim to create general-purpose representations enabling a variety of tasks, a comparison with specialized task-specific models is not possible due to task-specific model designs.

² <https://www.kaggle.com/competitions/pkdd-15-taxi-trip-time-prediction-ii>.

³ <https://iee-dataport.org/open-access/crawdad-epflmobility>.

⁴ <https://www.openstreetmap.org/>.

Table 1. Statistics of the road network and trajectory datasets.

	Road Network			Trajectory		
	#Intersections	#Road-Segments	Avg. Degree	#GPS-Points	#Trajectories	Coverage
Porto	5,358	11,331	2.4	74,269,739	1,544,234	97.9%
San Francisco	9,739	27,039	3.1	11,219,955	406,456	94.4%

4.3 Downstream Tasks and Evaluation Metrics

We consider four downstream tasks proposed in previous works [1, 16, 19]. For all downstream tasks, we pre-train the road representation models in an unsupervised manner and use the frozen embeddings to train a simple prediction model for each task. For **Label Classification (LC)** we select the road type as the label. For the road embedding models using the road type feature in the pre-training phase, we leave out that feature to evaluate prediction performance on unseen labels. We adopt a logistic regression classifier as the prediction model and report micro and macro F1 scores, denoted as Mi-F1 and Ma-F1. For **Traffic Inference (TI)**, we predict the average speed on the road segments. We adopt an MLP with a fully connected layer as the prediction model and report Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). For **Travel Time Estimation (TTE)** given a route, we input the sequence of road embeddings representing the route into a two-layer LSTM and predict the travel time of that route. For evaluation, we adopt RMSE and MAE. Finally, for the **Destination Prediction (DP)** task, we take the first 70% of the trajectory and input the corresponding sequence of road embeddings into a two-layer LSTM to predict the last visited location of the trajectory. We adopt the top-1 and top-5 prediction accuracy, denoted as ACC@1 and ACC@5.

4.4 Experimental Settings

We randomly selected 70% of the trajectory dataset for the representation learning. We used the remaining 30% for the training and evaluation of the trajectory-based downstream tasks **TTE** and **DP**. For those tasks, we further split the remaining trajectory set into 70% for training the prediction models and 30% for evaluation. For the road segment-based tasks **LC** and **TI**, we employed 5-fold cross-validation. We set the embedding dimension to 128 each for the SFC and SRE modules and employed the Adam optimizer with a learning rate of 0.001. For SRE, we used traffic volume for the traffic prediction task and set the weights $\lambda_{top} = \lambda_{struc} = \lambda_{trf} = \frac{1}{3}$. We set $l = 25$, $w = 5$, $n_{neg} = 3$ and performed 1000 walks per node. We trained SRE for ten epochs. For SFC, we set $k = 2$ and trained for 5000 epochs. We discuss parameter selection later in Sect. 4.7.

4.5 Performance Results

Table 2 summarizes the evaluation results for both datasets. As we can observe, our proposed TrajRNE approach consistently outperforms all the baselines on

Table 2. TrajRNE and baselines performance on two datasets and four tasks.

	Task	LC		TI		TTE		DP	
	Metric	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	ACC@1	ACC@5
Porto	GCN	0.660	0.411	14.175	20.361	77.490	110.589	0.246	0.502
	GAT	0.651	0.393	14.238	20.388	77.170	111.562	0.250	0.501
	RFN	0.498	0.087	15.088	20.908	83.125	116.160	0.246	0.492
	IRN2Vec	0.487	0.055	17.366	23.176	77.910	110.606	0.263	0.529
	HRNR	0.540	0.132	13.733	19.798	77.851	113.382	0.254	0.514
	Toast	0.440	0.206	13.793	19.543	78.807	111.270	0.265	0.534
	TrajRNE	0.682	0.496	13.228	19.215	75.495	109.067	0.270	0.546
San Francisco	GCN	0.663	0.070	10.435	16.294	118.832	273.111	0.027	0.071
	GAT	0.676	0.131	10.113	15.730	109.511	267.021	0.057	0.143
	RFN	0.672	0.125	9.819	15.660	113.185	266.541	0.042	0.112
	IRN2Vec	0.658	0.057	13.336	19.307	104.667	267.479	0.073	0.186
	HRNR	0.692	0.147	10.068	15.265	104.980	263.578	0.078	0.201
	Toast	0.662	0.068	10.122	15.631	109.057	265.514	0.046	0.121
	TrajRNE	0.759	0.475	8.437	13.334	97.776	256.812	0.097	0.238

both datasets and all tasks, demonstrating that incorporating trajectory information into road representation learning is essential for downstream application. Especially on the **LC** task, where the baselines predict only the most frequent labels, i.e., “residential”, with high accuracy, our TrajRNE approach outperforms the baselines by a large margin, in particular on the less frequent classes, as reflected by Ma-F1. It is worth noting that without learning road types explicitly, road embeddings created by TrajRNE enable us to predict the less frequent road types in the dataset with high precision. Comparing both datasets, San Francisco has many more road segments with fewer trajectory data, making the prediction for the most downstream tasks even more challenging. We observe that our approach outperforms the baselines on the San Francisco dataset by a larger margin. This result indicates that our TrajRNE approach can generate more robust road representations even with fewer trajectory data available.

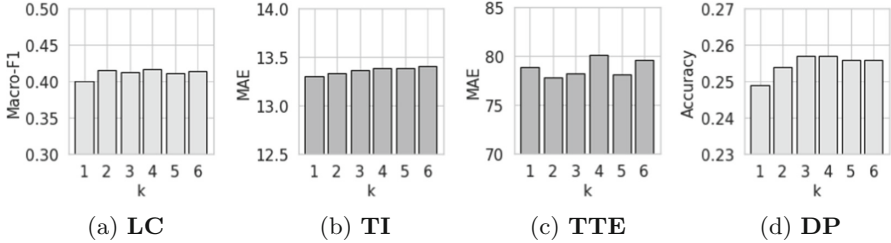
Regarding the baselines, we can observe that road embedding baselines mostly outperform graph representation methods, indicating that generic graph representation methods are unsuitable for road networks. Regarding the road representation baselines, Toast and HRNR outperform IRN2Vec and RFN in many cases, as the former utilize more specific road network-related information, e.g., extracting function zones or traveling semantics.

4.6 Ablation Study

To demonstrate the impact of the TrajRNE modules, we evaluate each module separately, i.e., TrajRNE(SFC) and TrajRNE(SRE). Table 3 presents the ablation study results. As we can observe, the modules indicate different strengths regarding specific tasks. While TrajRNE(SRE) outperforms TrajRNE(SFC) on

Table 3. Ablation study on different tasks on the Porto dataset.

Task	LC		TI		TTE		DP	
Metric	Mi-F1	Ma-F1	MAE	RMSE	MAE	RMSE	ACC@1	ACC@5
TrajRNE	0.682	0.496	13.228	19.215	75.495	109.067	0.270	0.546
TrajRNE(SFC)	0.689	0.458	13.353	19.719	77.449	113.828	0.255	0.515
TrajRNE(SRE)	0.511	0.254	13.692	19.549	76.609	109.599	0.269	0.544

**Fig. 3.** Impact of the k parameter of the TrajRNE(SFC) module for the Porto dataset on all tasks. Figures with light gray bars indicate the higher values are better, and for the dark gray bars, lower values are better.

the **TTE** and **DP** tasks, TrajRNE(SFC) achieves higher performance on the **LC** task. TrajRNE adopting both modules performs better than the modules isolated, except for the Mi-F1 score on the **LC** task. Regarding **LC**, the slight performance reduction of 1.0% for Mi-F1 is compensated by the increase in Ma-F1 by 8.3%. Overall, these results confirm that the TrajRNE modules provide complementary information and jointly provide the best performance.

4.7 Parameter Study

We examine the k parameter of the SFC module, which influences the receptive field of the method. Thus, with a higher k value, the module can observe a broader neighborhood. We evaluate the SFC module on all selected downstream tasks with varying k . The results are depicted in Fig. 3. We observe that the hyperparameter influence depends on the task. While for the **DP** task higher k value is better, for **TI**, lower k yields better performance. This is because, for **DP**, the distant neighborhood can be more important, as the destination will not typically be located in the local neighborhood. For traffic inference, the direct neighborhood contains the traffic most similar to the target road segment. We select $k = 2$ to balance across the downstream tasks.

5 Related Work

We discuss related work in road network representation and trajectory mining.

Road Network Representation Learning. As road networks are typically modeled as graphs, a natural way to learn representations is to use graph representation learning methods, e.g., GCN [9], and GAT [15]. RFN [7] adapted GCNs to road networks by proposing a relational fusion layer. IRN2Vec [16] used shortest paths to learn the geo-locality and was trained to predict road network tags. HRNR [19] extended graph convolutions by constructing a three-level hierarchical architecture to model road segments, functional, and structural zones. Toast [1] utilized the skip-gram model to learn the graph structure and refined the embeddings using a transformer-based model and an adapted pre-training objective. However, previous works relied on road network topology and static road features to learn road embeddings, which is insufficient for reflecting complex and dynamic traffic patterns and mobility behavior. To overcome these limitations, we extract traffic features and mobility behavior from vehicle trajectories and incorporate this information deeply into our model design. Thus, TrajRNE can learn the complex and dynamic behavior of road users observed in the network. We experimentally demonstrated that TrajRNE outperforms mentioned works on various downstream tasks.

Trajectory Mining. Vehicle trajectories are mined for many road network related tasks [4, 18], e.g., functional zones [14, 21], travel time on road networks [5] and next location prediction [2, 3, 11]. Some recent work incorporated trajectories into their model design for different tasks on road networks. Hong et al. [5] created a trajectory-based graph next to a road network graph to learn traffic behavior for travel time estimation jointly. Further, Li et al. [10] integrated flows from historical vehicle trajectories into their Trajectory-based Graph Neural Network model for traffic flow prediction. For short-term traffic speed prediction, Hui et al. [6] replaced the graph convolution networks by sampling trajectories and aggregating features along them. Inspired by Li et al. [10], we designed a graph convolution based on traffic flows and extended the idea by considering the traffic flow of the k-hop neighbors, thus increasing the receptive field.

6 Conclusion

In this paper, we presented TrajRNE – a novel road network representation learning approach incorporating information extracted from trajectories into its model design. TrajRNE comprises static road features, topology, traffic, and user mobility behavior. Specifically, we proposed the Spatial Flow Convolution, aggregating local and distant neighborhoods based on traffic flows. Further, we proposed the Structural Road Encoder, which learns the network topology, structure, and traffic, employing a multitask prediction objective. We incorporated user mobility behavior by weighting random walks with transition probabilities extracted from trajectories. We conducted extensive experiments on real-world datasets and evaluated TrajRNE against state-of-the-art road representation learning and graph representation methods. We demonstrated that TrajRNE consistently outperforms the baselines on four downstream tasks.

Acknowledgements. This work was partially funded by the DFG, German Research Foundation (“WorldKG”, 424985896), the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany (“d-E-mand”, 01ME19009B and “ATTENTION!”, 01MJ22012D), and DAAD, Germany (“KOALA”, 57600865).

References

1. Chen, Y., et al.: Robust road network representation learning: when traffic patterns meet traveling semantics. In: *CIKM 2021*, pp. 211–220. ACM (2021)
2. Dang, W., et al.: Predicting human mobility via graph convolutional dual-attentive networks. In: *WSDM (2022)*
3. Feng, J., et al.: Deepmove: predicting human mobility with attentional recurrent networks. In: *WWW (2018)*
4. Feng, Z., Zhu, Y.: A survey on trajectory data mining: techniques and applications. *IEEE Access* **4**, 2056–2067 (2016)
5. Hong, H., et al.: Heteta: heterogeneous information network embedding for estimating time of arrival. In: *KDD 2020*, pp. 2444–2454. ACM (2020)
6. Hui, B., Yan, D., Chen, H., Ku, W.: Trajnet: a trajectory-based deep learning model for traffic prediction. In: *KDD 2021*, pp. 716–724. ACM (2021)
7. Jepsen, T.S., Jensen, C.S., Nielsen, T.D.: Graph convolutional networks for road networks. In: *ACM SIGSPATIAL 2019*, pp. 460–463. ACM (2019)
8. Kipf, T.N., Welling, M.: Variational graph auto-encoders. *CoRR* abs/1611.07308 (2016)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *ICLR 2017*. OpenReview.net (2017)
10. Li, M., Tong, P., Li, M., Jin, Z., Huang, J., Hua, X.: Traffic flow prediction with vehicle trajectories. In: *AAAI 2021*, pp. 294–302. AAAI Press (2021)
11. Lin, Y., Wan, H., Guo, S., Lin, Y.: Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In: *AAAI 2021*, pp. 4241–4248. AAAI Press (2021)
12. Oono, K., Suzuki, T.: Graph neural networks exponentially lose expressive power for node classification. In: *ICLR 2020*. OpenReview.net (2020)
13. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: *ACM SIGKDD 2014*, pp. 701–710. ACM (2014)
14. Shimizu, T., Yabe, T., Tsubouchi, K.: Enabling finer grained place embeddings using spatial hierarchy from human mobility trajectories. In: *SIGSPATIAL (2020)*
15. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: *ICLR 2018*. OpenReview.net (2018)
16. Wang, M., Lee, W., Fu, T., Yu, G.: Learning embeddings of intersections on road networks. In: *SIGSPATIAL*, pp. 309–318. ACM (2019)
17. Wang, M., Lee, W., Fu, T., Yu, G.: On representation learning for road networks. *ACM Trans. Intell. Syst. Technol.* **12**(1), 11:1–11:27 (2021)
18. Wang, S., Bao, Z., Culpepper, J.S., Cong, G.: A survey on trajectory data management, analytics, and learning. *ACM Comput. Surv.* **54**(2), 39:1–39:36 (2022)
19. Wu, N., Zhao, W.X., Wang, J., Pan, D.: Learning effective road network representation with hierarchical graph neural networks. In: *KDD 2020*. ACM (2020)
20. Yang, C., Gidófalvi, G.: Fast map matching, an algorithm integrating hidden Markov model with precomputation. *Int. J. Geogr. Inf. Sci.* **32**(3), 547–570 (2018)
21. Yao, Z., Fu, Y., Liu, B., Hu, W., Xiong, H.: Representing urban functions through zone embedding with human mobility patterns. In: *IJCAI 2018* (2018)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

