Modeling Trajectories with Multi-task Learning

Kaijun Liu[1,2,4], Sijie Ruan[3,2], Qianxiong Xu[2], Cheng Long[2], Nan Xiao[4], Nan Hu[4], Liang Yu[4], Sinno Jialin Pan[2]
[1]*Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University (NTU), Singapore*
[2]*School of Computer Science and Engineering, NTU, Singapore* [3]*Xidian University, China* [4]*Alibaba Group, China*
kaijun001@e.ntu.edu.sg; sjruan@stu.xidian.edu.cn; {qianxion001@e., c.long@}ntu.edu.sg;
{nan.xiao, nan.h, liangyu.yl}@alibaba-inc.com; sinnopan@ntu.edu.sg

2022 23rd IEEE International Conference on Mobile Data Management (MDM) | 978-1-6654-5176-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/MDM55031.2022.00049

***Abstract*—With the increasing popularity of GPS modules, there are various urban applications relying on trajectory data modeling. In this work, we study the problem to model the vehicle trajectories by predicting the next road segment given a partial trajectory. Existing methods that model trajectories with Markov chain or recurrent neural network suffer from issues of modeling, context and semantics. In this paper, we propose a new trajectory modeling framework called Multi-task Modeling for Trajectories (MMTraj), which avoids these issues. Specifically, MMTraj uses multi-head self-attention networks for sequential modeling, captures the overall road network as the context information for road segment embedding, and performs an auxiliary task of predicting the trajectory destination to better guide the main trajectory modeling task (controlled by a carefully designed gating mechanism). Extensive experiments conducted on real-world datasets demonstrate the superiority of the proposed method over the baseline methods.**

***Index Terms*—Trajectory modeling; Road network; Multi-task learning;Transformer**

I. Introduction

With the increasing demand of location acquisition, GPS modules have been widely adopted and generated a huge set of trajectory data on road networks. Different from the trajectories that are generated by flights [1] or pedestrians [2] in free space, trajectories generated from road networks are constrained with road topological structure. These large-scale data based on road networks has empowered many downstream applications, such as urban vehicle navigation [3], popular route recommendation for drivers [4], travel time estimation [5] and so on. Trajectory modeling, which models the transition probabilities between two adjacent roads, is one of the fundamental problems on trajectory data and has received much attention in existing studies [6]–[8]. For example, it can be applied to the traffic simulation [9] to generate some paths for moving vehicles and simulate traffic conditions. It can also help the government and transportation agency to conduct traffic management and understand the overall traffic status in the near future to facilitate urban planning.

A few methods have been explored for the trajectory modeling problem. In [10], the authors use sample trajectories to pre-train a bi-gram sequence model for generating city-scale vehicular paths. In [7], the authors employ the first-order Markov model to forecast the most frequent action (make turning) at each road intersection for drivers' routing behaviour modeling. In [8], the authors explore Recurrent Neural Network (RNN) based models for trajectory modeling, which can deal with variable length trajectory sequences and capture the constraints of topological structure on road network. However, these existing methods still have limitations for the trajectory modeling problem, which we explain as follows.

Firstly, Markov chain or bi-gram model is known to be not sufficient to model the trajectory data that involves long-term dependencies [11]. RNN-based methods can only capture limited long-term dependencies and its inherently sequential learning nature precludes parallelization [12]. We call this the *modeling issue*. Secondly, while some existing methods such as the one in [8] considers the road network context (e.g., the method in [8] predicts the next road to be one of the neighboring roads of the current one), each road segment is still treated as individual token ID and the overall road network structure is not captured/utilized. We call this the *context issue*. Thirdly, a trajectory may have some semantics behind (e.g., it has an underlying yet unknown destination), but they are ignored by existing methods. We call this the *semantics issue*.

Understanding these limitations, we propose a new framework called Multi-task Modeling for Trajectories (MMTraj), which avoids the aforementioned issues of existing methods. Firstly, to mitigate the modeling issue, we adopt multi-head self-attention modules to encode the sequence information. Transformer with multi-head self-attention mechanism [12] has been proven to be more effective than RNN and adopted in Natural Language Processing (NLP) language modeling task [13]. Secondly, to solve the context issue, we use Graph Neural Network (GNN) [14] to learn the overall geometrical topology information of the road network and integrate the information into the road segment embeddings as the context information. Thirdly and most importantly, we model the a trajectory (i.e., to predict the next road segments) and predict the destination of the trajectory jointly with a multi-task learning strategy to solve the semantics issue. A specific gating mechanism is designed to control the confidence with which we trust the predicted destination information to help the main task of trajectory modeling.

In particular, we explain the intuition of using the predicted destination to guide the trajectory modeling with an example shown in Fig. 1. We have a known current trajectory of a vehicle (the black arrows), and it travels to the intersection A of the road segments. We aim to model the vehicle trajectory by predicting the next road segment that it would most probably travel and score the likelihood of all of its choices (1,2,3) at the intersection A. If we predict the road segment of the destination is the one with red flag as shown in Fig. 1,

2375-0324/22/$31.00 ©2022 IEEE
DOI 10.1109/MDM55031.2022.00049

there should be a higher probability for the driver to transit to road segment 1 instead of other road segments since the direction of this road segment is more towards the predicted destination. Therefore, it would be beneficial to predict the destination information as a guiding strategy to potentially enhance overall trajectory modeling.

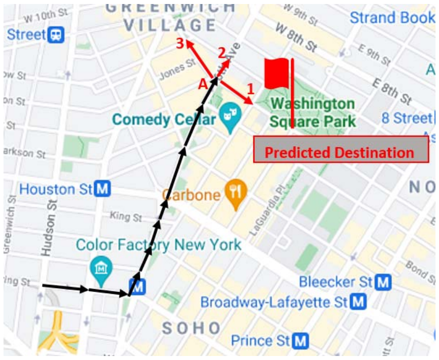

Fig. 1. Intuition of Trajectory Modeling with Destination Prediction

In summary, our major contributions in this work are listed as below:

- To the best of our knowledge, this is the first attempt to conduct trajectory modeling by predicting trajectory destination information as guidance via a multi-task learning strategy. A specific gating mechanism is designed for the multi-task learning to adjust the usage of destination information predicted and assist in the main task of trajectory modeling.
- A new framework is designed with GNN and dual-transformer architecture which utilize the overall geometrical graph topology of road network and multi-head self-attention mechanism in trajectory data sequence learning.
- We conduct extensive experiments on two real-world mobility datasets and demonstrate better performance of the proposed framework over the baseline methods.

II. Related Work

Trajectory modeling has been applied to many related problems on location based services. [15] proposes a hybrid prediction algorithm to predict the moving object's position in some delta time, which adopts different prediction algorithms according to the length of delta time. [16] adopts Markov decision process and inverse reinforcement learning to mine short-term and long-term evacuation behaviors for individuals. In [17], the authors design a system for finding efficient driving directions for a given destination, which benefits from trajectory modeling leveraging on intelligence of experienced drivers. [6] applies Bayesian Inverse Reinforcement Learning (BIRL) [18], which is based on first-order Markov chain, to model the transition probability. Most of these works apply first-order Markov chain to model the next road transition, however it is not suitable to capture the long-term dependencies and has limitations of sparsity problem [19]. More recently, [8] explores RNN for trajectory modeling, which can deal with the variable length trajectory sequences and capture the constraints of transitions (i.e., the next road segment can only be one of the adjacent road segments of the current one). In this paper, we follow the trajectory modeling setting in [8] and investigate an auxiliary task of predicting the trajectory destination to help with effective modeling. In addition, we adopt a self-attention based model for trajectory modeling.

III. Problem Formulation

A. Problem Definitions

The trajectory modeling problem was originally proposed in [8]. Here, we review the basic definitions and explain the problem formulation for trajectory modeling.

Definition 1 (Road Network). A road network can be represented as a directed graph $G = (\mathcal{V}, \mathcal{RS})$, where $\mathcal{V}$ is a set of vertices, $v \in \mathcal{V}$ represents an intersection between two road segments, $\mathcal{RS}$ is a set of edges, and each edge $rs \in \mathcal{RS}$ represents a road segment in road network.

Definition 2 (Route). A route $\mathbf{r} = [r_i]_{i=1}^{k}$ is a sequence of adjacent road segments, where r_i represents the i-th road segment, and k is the length of the route.

Definition 3 (Map-matched trajectory). A map-matched trajectory T is a sequence of road segments based on the underlying trajectory of a moving object after map matching. Note that a map-matched trajectory is always a route.

Problem Statement. Given a road network $G = (\mathcal{V}, \mathcal{RS})$ and a trajectory database $\mathcal{D} = \left\{T^{(i)}\right\}_{i=1}^{|\mathcal{D}|}$, we aim to obtain a model that receives i travelled road segments (route $r_{1:i}$) as input and predicts the next road segment r_{i+1}.

IV. Methodology

A. Framework Overview

We adopt the multi-task learning strategy for next road segment prediction (as the main task) and destination prediction (as an auxiliary task). The overview of the multi-task learning framework is shown in Fig. 2. The left side of the architecture aims to predict the destination information at each step as an auxiliary task. Then the predicted destination information (d_1 to d_{k-1}) is passed as part of inputs to right side of the architecture to predict the next road segment as the main task. There are some common components for the two tasks: Road Segment Embedding (RSE) Layer, Transformer Encoder and Fully Connected (FC) Layer.

For the auxiliary task, the input road segments (r_i) firstly pass through the RSE layer A and are embedded and processed as vector representations (embeddings) prepared for the sequential learning. These embeddings run through the Transformer Encoder A, decoded by a FC layer A and pass through a softmax function to predict the destination IDs (d_i) at each step. Next for the main task, the input sequences of road segment IDs (r_i) together with the predicted destination IDs (d_i) pass through RSE layer B. Normally the farther we travelled, with more confidence we would make use of the predicted destination information obtained in the auxiliary task. Therefore, instead of directly concatenating the input road segment embeddings and the predicted destination embeddings, a specific Multi-Task learning with Gating (MTG)

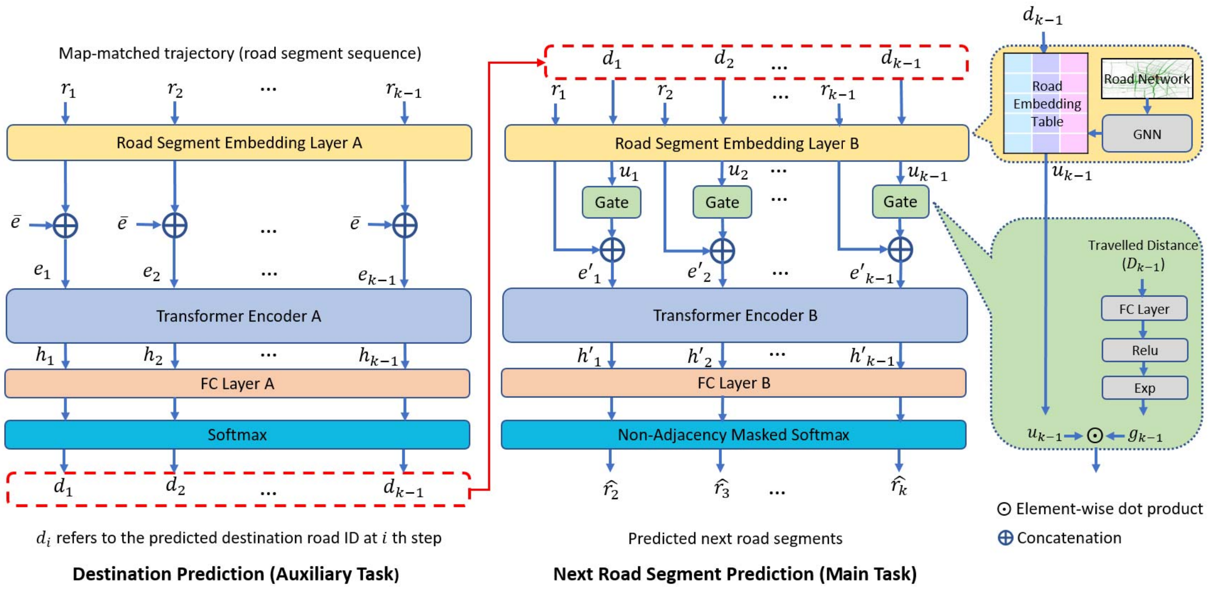
Map-matched trajectory (road segment sequence)
r_1 r_2 ··· r_{k-1}
Road Segment Embedding Layer A
$\bar{e}$ $\bar{e}$ ··· $\bar{e}$
e_1 e_2 ··· e_{k-1}
Transformer Encoder A
h_1 h_2 ··· h_{k-1}
FC Layer A
Softmax
d_1 d_2 ··· d_{k-1}
d_i refers to the predicted destination road ID at i th step
Destination Prediction (Auxiliary Task)
d_1 d_2 ··· d_{k-1}
r_1 r_2 ··· r_{k-1}
Road Segment Embedding Layer B
u_1 u_2 ··· u_{k-1}
Gate Gate ··· Gate
e'_1 e'_2 ··· e'_{k-1}
Transformer Encoder B
h'_1 h'_2 ··· h'_{k-1}
FC Layer B
Non-Adjacency Masked Softmax
$\hat{r}_2$ $\hat{r}_3$ ··· $\hat{r}_k$
Predicted next road segments
Next Road Segment Prediction (Main Task)
d_{k-1}
Road Embedding Table
Road Network
GNN
u_{k-1}
Travelled Distance (D_{k-1})
FC Layer
Relu
Exp
u_{k-1} ⊙ g_{k-1}
⊙ Element-wise dot product
⊕ Concatenation

Fig. 2. Framework Overview

mechanism is designed to control the usage of embedded predicted destination information (u_i) from the auxiliary task to the main task. Then input road segment embeddings and the predicted destination embeddings after the gate are concatenated, passed through the Transformer Encoder B for sequential learning and decoded by FC layer B. Finally, a non-adjacency masked softmax is utilized [20] for the output of the FC layer B and helps mask out the non-adjacent roads during predicting for final next road segment. Next, we present the key modules of this framework in detail.

B. Road Segment Embedding (RSE) Layer

The first module is to learn the input road segment embeddings. In this paper, we propose to leverage the Graph Neural Networks (GNN) techniques by considering the whole road network as a geometrical graph and learn node embeddings (road segment embeddings). We utilize the road network topology information, transform each road segment rs to a node in the graph and formulate the adjacency matrix of our road network graph. We take the road segment IDs, embed them as one hot vectors and feed them to the GNN. Since we have two different tasks (auxiliary task and main task), we design two separate Road Segment Embedding (RSE) Layers to learn the embeddings.

The zoom-in details view of the Road Segment Embedding (RSE) Layer is as shown at upper-right side of Fig. 2. We apply an existing GNN method GCN [21] on the road network graph to transform an one-hot representation of a node (which represents a road segment) to a low dimensional feature embedding. All node feature embeddings together will form a trainable Road Embedding Table (with dimension $EmbeddingSize \cdot |\mathcal{RS}|$). Then we perform one of pooling strategies (mean pooling) on Road Embedding Table here to obtain a graph embedding $\bar{\mathbf{e}}$ [22]. The input road segments (r_1 to r_{k-1}) will pass the RSE layer and query out the respective road segment embedding vectors directly from the trainable Road Embedding Table. In particular, for the road segment embedding layer A, the graph embedding $\bar{\mathbf{e}}$ obtained from GNN is concatenated with the individual queried input road segment embedding and their concatenation output serves as the final input ($\mathbf{e_i}$) into the next module Transformer Encoder A. Our road segment embedding layer differs from existing ones [23]–[25] in that it combines the road network embedding with the road segment embeddings to better capture the overall topological information of the road network.

C. Transformer Encoder

The road segment embeddings obtained from the RSE layer are applied as part of inputs to transformer encoder [12]. To avoid the loss of sequential information of trajectory, we also add the popular sinusoidal positional encoding $\mathbf{p}_i$ into the input embeddings $\mathbf{e}_i$ as the final input $\mathbf{x}_i = \mathbf{e}_i + \mathbf{p}_i$. We adopt Scaled Dot-Product Attention, which can be treated as mapping a query (Q) and a set of key-values (K, V) pairs to an output vector. The output vector is a weighted sum of values (V), and the weights can be calculated using query and key through a compatibility function. Mathematically, it is defined as

$$\mathrm{Attention}\ (\mathrm{Q}, \mathrm{K}, \mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{QK}^T}{\sqrt{dim_k}}\right)\mathrm{V} \tag{1}$$

Furthermore, we use multi-head self-attention to enhance the learning of road segment sequence. Given the input representations $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_{k-1}}]$, the final output of multi-head attention is marked as Z, where W is the self-attention learning parameters:

$$\mathrm{Z} = \mathrm{MHAttn}(\mathrm{X}) = [\mathrm{head}_1, \ldots, \mathrm{head}_h] \cdot \mathrm{W}^O \tag{2}$$

$$\text{head}_i = \text{Attention}\left(\mathrm{XW}_i^Q, \mathrm{XW}_i^K, \mathrm{XW}_i^V\right) \tag{3}$$

We also follow [12] with the setting of position-wise feed-forward network, residual connection and layer normalization [26]. The final output of the transformer encoder is denoted as $\mathbf{H}/\mathbf{H}'$ (separated with vector $\mathbf{h_i}/\mathbf{h'_i}$ at the i-th step as shown in the Fig. 2). Here we design two transformer encoders for sequence learning since we have two different tasks and prediction targets.

D. Multi-Task Learning with Gating (MTG) Mechanism

As mentioned in the framework overview section, multi-task learning is adopted to jointly train for the next road segment prediction process while predicting the destination information for guidance at the same time. The transformer encoder A encodes the road segment embedding sequence information, which then is passed through a fully connected layer for predicting the destination ID at each step. The predicted destination IDs together with the input road segment sequence are both embedded through the Road Segment Embedding layer B.

In the main task, a gating mechanism is specifically designed for controlling the confidence and usage of predicted destination information instead of concatenating the two input embeddings directly. The idea behind is that it tends to be more difficult to predict the destination ID when we have traveled a shorter distance only. The zoom-in details view of the gating mechanism is provided at the lower-right side of Fig. 2. Instead of designing a relational polynomial function to address the relationship between travelled distance and gating confidence, we adopt a linear neural network (FC layer) to learn this. Mathematically, the travelled distance (D) is calculated by the summation of the lengths of travelled road segments. We apply the exponential decay function to simulate the natural decay relationship and its output is also a variable between 0 and 1 to represent a confidence value. The gating function g is designed as shown in the equation below. The gating g_i will perform element-wise dot product with the embedding u_i (as shown in Fig. 2).

$$g_i = \exp\left(-\max\left(0, \mathbf{W}_g\left[D_i\right] + \mathbf{b}_g\right)\right) \tag{4}$$

E. Non-Adjacency Masked Softmax

A non-adjacency masked softmax is utilized [20] for the final fully connected layer to predict the next road segment. This is to capture the road network topology and adjacency relationship so that the next road predicted is constrained within the road adjacency list instead of the whole road segment set $\mathcal{RS}$.

$$P(\hat{r}_{i+1} = j|\mathbf{h}_i) = \frac{\exp(\mathbf{h}_i^\top \cdot \mathbf{w}_j) \cdot M_{r_i,j}}{\sum_{p \in RS} \exp(\mathbf{h}_i^\top \cdot \mathbf{w}_p) \cdot M_{r_i,p}} \tag{5}$$

where $\mathbf{h}_i$ is the output of the transformer at step i, $\mathcal{RS}$ is the set of all road segments, $\{\mathbf{w}_j|\forall j \in \mathcal{RS}\}$ is j-th column vector from a trainable parameter matrix $\mathbf{W}$ of the fully connected layer, $M_{r_i} \in \{0,1\}^{|RS|}$ is the non-adjacency road segment mask of road segment r_i.

F. Training

The training of our model is end to end. During the training phase, the model predicts destination ID (auxiliary task) and next road segment ID (main task) simultaneously. The loss functions of two tasks are defined as $\mathcal{L}_{dest}$ and $\mathcal{L}_{main}$ below based on cross-entropy loss. Specifically, the auxiliary loss $\mathcal{L}_{dest}$ is calculated by cross-entropy using the last road segment at drop off point as the ground-truth destination ID. Overall, the final model optimization function is weighted combination of these two loss functions and formulated as:

$$\mathcal{L}(\theta) = \mathcal{L}_{main}(\theta) + \lambda\mathcal{L}_{dest}(\theta) \tag{6}$$

where λ is the parameter of multi-task loss weight and θ denotes the parameters of the model. The model is trained to minimize the loss $\mathcal{L}(\theta)$.

V. Experiments

A. Experiment setting

1) Datasets: We use two real-world taxi trajectory datasets and the road network from OpenStreetMap [1] to validate the effectiveness of our model. The trajectories of two cities, Chengdu and Xi'an are obtained from public datasets released by DiDi Chuxing [2]. There are 2,761 (5,403 for Xi'an dataset) road segments in the Chengdu selected area and total sample of 696,477 trajectories for Chengdu dataset (752,849 for Xi'an dataset). We split the dataset into training set, validation set and test set with a splitting ratio of 8:1:1. We perform map matching [27] process on the raw GPS trajectories and map them to the sequences of road segments for preparation of our task.

2) Evaluation Metrics: Similar to the previous work [8], the negative log-likelihood (NLL) and prediction accuracy (ACC) are adopted as the evaluation metric. ACC calculates the total prediction accuracy of next road segments by the road with the maximum predicted probability. The below equations are formulated for a test set with N trajectories,

$$NLL = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k_i-1}\log P\left(r_{j+1} \mid r_{1:j}\right) \tag{7}$$

$$ACC = \frac{1}{\sum_{i=1}^{N} k_i}\sum_{j=1}^{k_i-1} 1\left\{\operatorname{argmax}_{r \in RS} P\left(r \mid r_{1:j}\right) = r_{j+1}\right\} \tag{8}$$

B. Baselines

- **HA (Historical Average)** is a common statistical method to calculate the average transition rate of the two adjacent road segments based on a historical trajectory database. We predict the next road segment by choosing the one with highest historical transition rate.
- **N-gram** [28]. The traditional methods using the second and third-order Markov chains serve as basic baseline, and the conventional name N-gram is used to label them.

[1] http://www.openstreetmap.org

[2] https://outreach.didichuxing.com

TABLE I
MODEL PERFORMANCE COMPARISON

Methods	Chengdu		Xi'an	
	ACC	NLL	ACC	NLL
HA	83.93%	8.02	84.41%	8.57
Tri-gram	85.48%	7.00	85.58%	8.08
4-gram	86.28%	6.47	86.18%	7.81
RNN	87.97%	6.27	87.82%	7.10
CSSRNN	88.01%	6.17	87.83%	6.95
LPIRNN	88.02%	6.15	87.85%	6.94
MMTraj – MTG&RSE	88.14%	6.22	87.87%	7.03
MMTraj – MTG	88.17%	6.17	87.90%	6.98
MMTraj – RSE	90.55%	5.23	90.04%	5.94
MMTraj – Transformer	90.60%	5.12	90.06%	5.79
MMTraj	**90.66%**	**5.05**	**90.17%**	**5.77**

Following the same setting, we adopt Laplace smoothing for sparsity [29] which only smooths legal transitions.

- **RNN** [30] is a popular deep learning based method to build the sequence model directly adopting traditional RNN. We apply LSTM with the same embedding size as our model MMTraj uses.
- **CSSRNN** [8] is the state-of-art model for the trajectory modeling setting. The method is an extension of traditional RNN and addresses the issue of topological constraints.
- **LPIRNN** [8] is another high-performance variant related to CSSRNN and it incorporates the topological information externally and performs the prediction by multiple individual tasks.

C. Variants for Ablation Study

We also implement and compare MMTraj with different variants of our model for the ablation study later.

- **MMTraj-MTG&RSE**: The basic model we implement with a single transformer encoder [12] using non-adjacency masked softmax for next road segment prediction. We remove the multi-task learning with gating (MTG) mechanism and the road segment embedding (RSE) parts.
- **MMTraj-MTG**: We remove multi-task learning with gating (MTG) mechanism from our model MMTraj.
- **MMTraj-RSE**: We replace the road segment embedding (RSE) layer with a traditional FC embedding layer.
- **MMTraj-Transformer**: We replace the dual transformer encoder with dual LSTM.

D. Results

*1) **Overall Performance**:* In this section, we compare our model with the baselines over the two real world taxi trajectory datasets. The performance accuracy (ACC) of different approaches for next road prediction is presented in Table I. It can be observed that our model MMTraj clearly outperforms all the baselines and variants over both datasets.

First, Historical Average (HA) method, which is a statistical method to calculate the average transition rate of any two adjacent road segments based on historical data, provides a fundamental performance range (around 83% to 84%) of these two datasets. Markov chain based models (Tri-gram, 4-gram) perform better than HA because they are learned as a model based method instead of purely considering historical statistics. For RNN based models, LPIRNN seems to perform slightly better than the other RNN variants. As mentioned in [8], for datasets with high density, LPIRNN may perform better than CSSRNN.

It is shown that our full model (MMTraj) obtains the best performance across all of the evaluation metrics. Our approach outperforms the current best baseline LPIRNN by a delta value 2.64% on Chengdu dataset and 2.32% on Xi'an dataset. It also achieves the lowest NLL among all the methods across both datasets. The fundamental reasons for such improvement mainly consist of two aspects. First, the main reason is that we design a multi-task learning framework with gating mechanism for modeling trajectory meantime predicting destination as an auxiliary task for guidance. Second, we study some key components such as Road Segment Embedding (RSE) layer and transformer encoder to enhance the learning capacity further. We will perform the ablation study in next section.

*2) **Ablation Study**:* We try to elaborate some ablation analysis to understand some impacts of key components on model performance.

MMTraj-MTG&RSE is the basic version we implement using a single transformer encoder with non-adjacency masked softmax for prediction. The overall module ACC performance drops much (2.52% and 2.3%) respectively on two datasets comparing with our full model MMTraj. It demonstrates that MTG and RSE module together make main contribution in the model performance improvement. The second variant (MMTraj-MTG) removing multi-task learning with gating (MTG) mechanism has the ACC performance drop 2.49% and 2.27% respectively comparing with the full model MMTraj on the Chengdu and Xi'an datasets It shows that the multi-task design with gating plays a significant role in our full model design. It also shows that our idea of predicting the destination as guidance works as expected and using gating mechanism to control the usage of predicted destination information assists in the next road segment prediction.

The third variant (MMTraj-RSE) removing road segment embedding (RSE) layer shows around 0.1% ACC drop comparing with the full model. Possibly it is due to the road network adjacency constraints are already considered during decoding through our non-adjacency masked softmax function. Based on the observation, RSE can still consistently provide a certain of further improvement over other baselines and this is also indispensable especially when there are huge number of road transitions involved in all trajectories. The fourth variant (MMTraj-Transformer) is that we replace dual transformer encoder with dual LSTM and keep MTG and RSE the same. The overall ACC performance drops a bit over the two datasets comparing with our full model. However, the overall ACC performance is still higher than the best baseline by 2.58% and 2.21% respectively on two datasets. It can be inferred that the idea and overall framework of multi-task learning with gating mechanism make main contribution in the model performance improvement while RSE and transformer encoder act as facilitators to further improve the performance.

VI. Conclusion

In this paper, we propose a new end-to-end Multi-task learning model MMTraj for trajectory modeling. MMTraj adopts multi-head self-attention networks for sequential modeling, captures the overall road network via GNN as the context information for road segment embedding, and performs an auxiliary task of predicting the trajectory destinations to help with the main trajectory modeling task (controlled by a carefully designed gating mechanism). Extensive experiments are conducted on two real-world mobility datasets and demonstrate the superiority of the proposed framework (over 2.3% improvement) compared with the baseline methods. In the future, we plan to explore other settings of trajectory modeling (e.g., with more context information including the user, time and external factors available).

Acknowledgments

This research is supported in part by Alibaba Group through Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore, and by the Ministry of Education, Singapore, under its Academic Research Fund (Tier 2 Award MOE-T2EP20220-0011). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore. We thank all reviewers for their constructive suggestions for improving this paper.

References

[1] H. Georgiou, N. Pelekis, S. Sideridis, D. Scarlatti, and Y. Theodoridis, "Semantic-aware aircraft trajectory prediction using flight plans," *International Journal of Data Science and Analytics*, vol. 9, no. 2, pp. 215–228, 2020.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[3] R. R. Joshi, "A new approach to map matching for in-vehicle navigation systems: the rotational variation metric," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*. IEEE, 2001, pp. 33–38.

[4] G. Cui, J. Luo, and X. Wang, "Personalized travel route recommendation using collaborative filtering based on gps trajectories," *International journal of digital earth*, vol. 11, no. 3, pp. 284–307, 2018.

[5] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2697–2705.

[6] J. Zheng and L. M. Ni, "Modeling heterogeneous routing decisions in trajectories for driving experience learning," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 951–961.

[7] N. Banovic, T. Buzali, F. Chevalier, J. Mankoff, and A. K. Dey, "Modeling and understanding human routine behavior," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 248–260.

[8] H. Wu, Z. Chen, W. Sun, B. Zheng, and W. Wang, "Modeling trajectories with recurrent neural networks," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3083–3090. [Online]. Available: https://doi.org/10.24963/ijcai.2017/430

[9] L. Li, R. Jiang, Z. He, X. M. Chen, and X. Zhou, "Trajectory data-based traffic flow studies: A revisit," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 225–240, 2020.

[10] N. Xiao, N. Hu, L. Yu, and C. Long, "Generating full spatiotemporal vehicular paths: A data fusion approach," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2837–2844.

[11] M. Srivatsa, R. Ganti, J. Wang, and V. Kolar, "Map matching: Facts and myths," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 484–487.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[14] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[15] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou, "A hybrid prediction model for moving objects," in *2008 IEEE 24th international conference on data engineering*. IEEE, 2008, pp. 70–79.

[16] X. Song, Q. Zhang, Y. Sekimoto, T. Horanont, S. Ueyama, and R. Shibasaki, "Modeling and probabilistic reasoning of population evacuation during large-scale disaster," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1231–1239.

[17] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 220–232, 2011.

[18] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning." in *IJCAI*, vol. 7, 2007, pp. 2586–2591.

[19] H. Wu, J. Mao, W. Sun, B. Zheng, H. Zhang, Z. Chen, and W. Wang, "Probabilistic robust route recovery with spatio-temporal dynamics," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1915–1924.

[20] H. Ren, S. Ruan, Y. Li, J. Bao, C. Meng, R. Li, and Y. Zheng, "Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1410–1419.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[22] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, and V. Sheinin, "Graph2seq: Graph to sequence learning with attention-based neural networks," *arXiv preprint arXiv:1804.00823*, 2018.

[23] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Graph convolutional networks for road networks," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 460–463.

[24] Y. Chen, X. Li, G. Cong, Z. Bao, C. Long, Y. Liu, A. K. Chandran, and R. Ellison, "Robust road network representation learning: When traffic patterns meet traveling semantics," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 211–220.

[25] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 6–14.

[26] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[27] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.

[28] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[29] D. M. Christopher, R. Prabhakar, and S. Hinrich, "Introduction to information retrieval," 2008.

[30] A. Graves, "Generating sequences with recurrent neural networks," *ArXiv*, vol. abs/1308.0850, 2013.