Empowering Next POI Recommendation with Multi-Relational Modeling

Zheng Huang
University of Virginia
Charlottesville, Virginia, USA
zh4zn@virginia.edu

Jing Ma
University of Virginia
Charlottesville, Virginia, USA
jm3mr@virginia.edu

Yushun Dong
University of Virginia
Charlottesville, Virginia, USA
yd6eb@virginia.edu

Natasha Zhang Foutz
University of Virginia
Charlottesville, Virginia, USA
nfoutz@virginia.edu

Jundong Li
University of Virginia
Charlottesville, Virginia, USA
jl6qk@virginia.edu

ABSTRACT

With the wide adoption of mobile devices and web applications, location-based social networks (LBSNs) offer large-scale individual-level location-related activities and experiences. Next point-of-interest (POI) recommendation is one of the most important tasks in LBSNs, aiming to make personalized recommendations of next suitable locations to users by discovering preferences from users' historical activities. Noticeably, LBSNs have offered unparalleled access to abundant heterogeneous relational information about users and POIs (including user-user social relations, such as families or colleagues; and user-POI visiting relations). Such relational information holds great potential to facilitate the next POI recommendation. However, most existing methods either focus on merely the user-POI visits, or handle different relations based on over-simplified assumptions while neglecting relational heterogeneities. To fill these critical voids, we propose a novel framework, MEMO, which effectively utilizes the heterogeneous relations with a multi-network representation learning module, and explicitly incorporates the inter-temporal user-POI mutual influence with the coupled recurrent neural networks. Extensive experiments on real-world LBSN data validate the superiority of our framework over the state-of-the-art next POI recommendation methods.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

next POI recommendation; location prediction; multi-relations

1 INTRODUCTION

The era of information explosion has witnessed rapid developments and adoptions of mobile devices and web applications. As a result, location-based social networks (LBSNs) have recorded rich activities at different *points-of-interests (POIs)*, such as shopping malls, restaurants, or gyms. Next POI recommendation represents one of the most important tasks in LBSNs [12, 27]. It aims to suggest the next suitable location(s) for each person based on his/her preference that can be learned from his/her prior information on LBSNs, such as user profile, social relationship, and historical activities.

Prior research has performed next POI recommendation by capturing users' preferences from their trajectories, i.e., users' temporally ordered location information at different POIs. For instance, sequential statistical models (e.g. Markov chain) [2, 4] are widely used. In addition, graph-based methods [6, 24] learn both user and POI representations for recommendation by modeling each user-POI visit as an edge in a graph. Another line of research [3, 27, 28] applies recurrent neural networks (RNNs) based models to capture users' preferences by modeling users' trajectories as sequences. More recent studies adopt self-attention [22] to capture long-term dependencies and spatio-temporal correlations among POIs [13, 15].

Despite the success of these methods, next POI recommendation on LBSNs remains a daunting task due to the following challenges: 1) users' preferences are complex and often bear strong connections with different types of social relations [18, 20]. As indicated in [16], different social relations affect users differently. For example, users seek recommendations on shopping malls from family members, whereas recommendations on training institutions from colleagues (Fig. 1). These heterogeneous social relations, together with the user-POI relations formed by the users' visits to POIs, bring great challenges to effectively utilize the rich information embedded. Most existing methods [14, 19, 24, 25] either focus solely on the user-POI relations, or assume the node representations in different relation-specific networks are within the same latent space. 2) Capturing the crucial, inter-temporal, mutual influence between the users and POIs remains challenging. In the user-POI relations, users' preferences and POIs' latent status (e.g., reputation) reciprocally influence each other over time. As shown in Fig. 1, if Anna left a positive comment about a shopping mall that she recently visited,
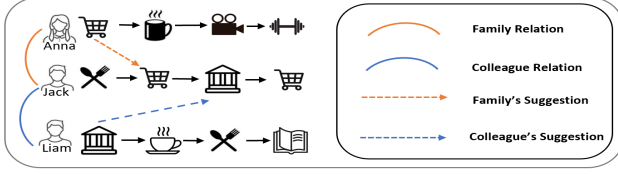
Figure 1: Heterogeneous social relationship: Jack takes sister Anna's recommendations on shopping malls and colleague Liam's recommendations on training institutions.

or recommended it to others, then the reputation and popularity of this POI may be elevated. In turn, the experience and reputation with this POI will affect Anna's future visitation.

To address these challenges, we propose a novel framework, Multi-Relational Modeling (MEMO) for next POI recommendation. First, to utilize the diverse user-user social relations (e.g., families or colleagues) and user-POI relations, we develop a multi-relational modeling module based on multiple graph convolutional networks (GCNs) [5] and self-attention [9]. Specifically, we map each type of relation to a corresponding network to learn the relation-specific representations; and then adopt a self-attention mechanism to aggregate the user representations from the different relation types. Next, to capture the mutual influence between the users and POIs over time, we design a user-POI mutual influence modeling component based on coupled recurrent neural networks (RNNs) that mutually update each other's representations [10].

In summary, this research contributes to the literature along three dimensions. One, it extends the extant studies on next POI recommendation by accounting for the critical, yet overlooked, heterogeneous relations, particularly heterogeneous social relations among users, and inter-temporal user-POI mutual influence. Two, this research proposes a novel framework, MEMO, to incorporate the heterogeneous relations with a multi-network representation learning module, and to capture the inter-temporal user-POI mutual influence with the coupled RNNs. Three, it substantiates the proposed framework via the experiments on large-scale real-world data and comparisons with the state-of-the-art baselines.

2 PROBLEM STATEMENT

Let $\mathcal{U} = \{u_1, u_2, ..., u_U\}$ be a set of U users, and $\mathcal{L} = \{l_1, l_2, ..., l_L\}$ be a set of L POIs. Each POI $l \in \mathcal{L}$ is associated with a geo-coordinate (longitude, latitude) indicating its geographical location.

Trajectory. The trajectory of each user $u_i \in \mathcal{U}$ is a time-ordered sequence: $C(u_i) = \{c_1(u_i), c_2(u_i), ..., c_K(u_i)\}$. Each element $c_k(u_i)$ can be represented with a tuple (l_k, t_k), where l_k is the geo-coordinate of the POI visited by user u at timestamp t_k.

User-User Social Relations. Let $\mathcal{R} = \{r_1, r_2, ..., r_P\}$ denote a set of P social relations among the users. We define a user-user social relation as a triple (u_i, u_j, r_p), where $u_i, u_j \in \mathcal{U}$, $r_p \in \mathcal{R}$, indicating that user u_i and u_j have a social relation of type r_p, which is assumed to be symmetric for simplicity. Note that there may exist multiple types of social relations between two users.

Next POI Recommendation. At each timestamp t, the next POI recommendation task takes each user's trajectory from timestamps 1 to t and different types of social relations as the input, and recommends a list of the suitable POIs with the highest prediction scores for each user to visit at next timestamp $t + 1$.

3 THE PROPOSED FRAMEWORK

We now introduce the proposed framework, MEMO, consisting of two main components (Fig. 2): 1) a **relation modeling** module to effectively utilize the heterogeneous relations, and 2) a **user-POI mutual influence modeling** module to capture the mutual influence between the users and POIs over time.

3.1 Relation Modeling

To capture each user's preferences for the next POIs, MEMO leverages the heterogeneous relations, including multiple user-user social relations and user-POI relations collected from the LBSNs. Specifically, to model P different types of user-user social relations among the user set $\mathcal{U}$, we enlist P networks $\mathcal{G}_1, ..., \mathcal{G}_P$, each with its corresponding symmetric adjacency matrix $\mathbf{A}_1, ..., \mathbf{A}_P$, where $\mathbf{A}_p \in \mathbb{R}^{U \times U}$ and $\mathbf{A}_p[i, j] = \mathbf{A}_p[j, i] = 1$ if there is a type p social relation between user u_i and u_j. Similarly, the user-POI network $\mathcal{G}_C$ corresponds to an adjacency matrix $\mathbf{A}_C \in \mathbb{R}^{(U+L) \times (U+L)}$ that captures whether a user has visited a specific POI or not. In total, there are $P + 1$ types of relations, including P types of user-user social relations and 1 type of user-POI relations.

Relation-Specific Representation Learning. To accommodate the diverse types of relations, we leverage a relation-specific representation learning module to map the nodes into a latent representation space corresponding to each relation separately. First, for each node v_i, a general embedding $\mathbf{x}_i$ is initialized randomly. We then use a relation-specific transition function $\Phi_p(\cdot)$ to map each $\mathbf{x}_i$ to a new embedding $\mathbf{x}_i^p$ corresponding to the p-th relation type for any $p \in [P + 1]$: $\mathbf{x}_i^p = \Phi_p(\mathbf{x}_i)$. Based on $\mathbf{x}_i^p$, we learn a relation-specific representation $\mathbf{h}_i^p$ for each node v_i by aggregating the embeddings of the neighbors on each network $\mathcal{G}_p$, which is implemented by a GCN [5] layer with an attention mechanism [23, 26].

Aggregation over Different Relation Types. As discussed in [8], one should not simply assume homogeneous relations or nodes representations across the relation-specific networks within the same latent space. Inspired by [8], we hence utilize a self-attention mechanism [22] to aggregate all relation-specific representations of each node into a common latent space. This allows us to effectively capture each user's preference embedded in the different types of relations. Specifically, given each node v_i, and all its representations in the different networks, we map each representation corresponding to the p-th ($p \in [P + 1]$) relation type $\mathbf{h}_i^p$ into a Key vector $\mathbf{k}_i^p = \mathbf{W}_p^K \mathbf{h}_i^p$, a Query vector $\mathbf{q}_i^p = \mathbf{W}_p^Q \mathbf{h}_i^p$, and a Message vector $\mathbf{m}_i^p = \mathbf{W}_p^M \mathbf{h}_i^p$, respectively. Here $\mathbf{W}_p^Q, \mathbf{W}_p^K, \mathbf{W}_p^M$ are the trainable parameters used for the above linear projections. We then calculate the attention weight between each pair of relation types (p_1, p_2) with the similarity between the p_1-th Query vector and the p_2-th Key vector: $\alpha(p_1, p_2) = \text{Softmax}_{\forall p_2 \in [P+1]}(\mathbf{k}_i^{p_2} \mathbf{q}_i^{p_1} / \sqrt{d})$, where d is the representation dimension. Note that different from the ordinary attention mechanisms [23] that assume the same latent space for the input representations, the above linear projections in our framework are relation-specific, hence allowing us to aggregate the relation-specific representations from different latent spaces. We then aggregate the representations from the different relation types as $\mathbf{h}_i = \text{MLP}([\tilde{\mathbf{h}}_i^1 || \tilde{\mathbf{h}}_i^2 || ... || \tilde{\mathbf{h}}_i^{P+1}])$, where $||$ is a concatenation operation, $\text{MLP}(\cdot)$ is a multilayer perceptron, and
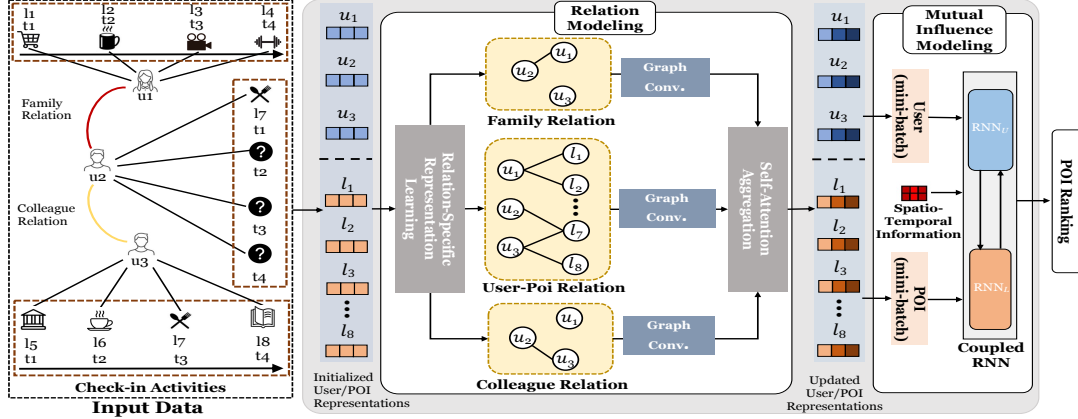
Figure 2: Overview of *MEMO* for next POI recommendation.

Table 1: Summary statistics of datasets.

	#Users	#POIs	#Visits	#Family Relations	#Colleague Relations
B'more July	17,892	559	1,753,382	19,026	17,646
DC July	7,517	4,213	262,679	1,100	12,316
DC August	3,418	3,450	277,093	1,290	14,250

$\tilde{\mathbf{h}}_i^p = \oplus_{p' \in [P+1]} \{\alpha(p, p') \cdot \mathbf{m}_i^{p'}\}$ is the representation that aggregates the messages from all relation types to the p-th relation type, with $\oplus$ denoting the element-wise sum.

3.2 User-POI Mutual Influence Modeling

In the user-POI relations, the users' and POIs' latent status may be mutually influenced by each other over time. We hence need to update the users' and POIs' representations to capture such a mutual influence. Specifically, we deploy the coupled RNNs [10] consisting of a user RNN (RNN_U) and a location RNN (RNN_L) to respectively learn the users' and POIs' representations at each timestamp.

Here, RNN_U incorporates the POIs' representations to update the users' representations, and vice versa. Suppose that user u visited POI l at timestamp t in $C(u)$, user u's representation at timestamp $t + 1$ is influenced by location l's representation at timestamp t, formally derived as: $\mathbf{h}_u^{t+1} = \sigma(\mathbf{W}_1^U \mathbf{h}_u^t + \mathbf{W}_2^U \mathbf{h}_l^t + \mathbf{W}_3^U \mathbf{z}^{\triangle t} + \mathbf{W}_4^U \mathbf{z}^{\triangle d})$. The matrices $\mathbf{W}_1^U, ..., \mathbf{W}_4^U$ are the parameters of RNN_U. For simplicity, we use $\mathbf{h}_u^{t+1}$ and $\mathbf{h}_u^t$ to respectively denote user u's representations at timestamp $t + 1$ and t; and $\mathbf{h}_l^t$ the representation of location l at timestamp t. Here, $\mathbf{z}^{\triangle d}$ and $\mathbf{z}^{\triangle t}$ are the representations incorporating the spatio-temporal information to facilitate next POI recommendation (details later in this section).

Similarly, RNN_L also leverages the users' representations to update the representations of each POI l at next timestamp $\mathbf{h}_l^{t+1}$. Specifically, we denote the list of users who visited l at timestamp t as $\mathcal{U}(l, t)$, and denote the list of their representations as $\mathbf{h}_{\mathcal{U}(l,t)}^t$. The representation of POI l is updated as: $\mathbf{h}_l^{t+1} = \text{RNN}_L(\mathbf{h}_l^t, \mathbf{h}_{\mathcal{U}(l,t)}^t)$. At each timestamp t, the representation of POI l is recursively updated $|\mathcal{U}(l, t)|$ times in the chronological order of the users' visits. To update the user-POI representations in parallel while maintaining the time dependency, we create each mini-batch by selecting the independent user-POI visits from the users' trajectories, to ensure that every two visits processed in the same batch do not share any common users or POIs [10].

Table 2: Comparison of next POI recommendation methods.

|	Model	User Embeddings	Social Relations	Trajectories
Statistical Method	Markov [17]			✓
Graph Network	LightGCN [6]	✓	✓	
Recurrent Network	GRU4Rec [7] DeepMove [3]			✓ ✓
Self-attentive Method	SASRec [9] TiSASRec [11] STAN [15]	✓		✓ ✓ ✓
Our Method	**MEMO**	✓	✓	✓

Spatio-Temporal Representation. Apart from the users' relations and trajectories, we also incorporate the spatio-temporal information to facilitate next POI recommendation. For instance, when roaming around, a user might be interested in visiting a club near his/her current location (spatial) within the next few hours (temporal). We hence design a mechanism to efficiently incorporate such spatio-temporal information by utilizing the spatial transition (distance) and time transition (visiting time intervals) between each pair of POIs consecutively visited by a user. Specifically, we introduce two learnable vectors, $\mathbf{t}_s$ and $\mathbf{t}_l$ to respectively represent the short and long time transitions. The representation of a time interval is then formally obtained as: $\mathbf{z}^{\triangle t} = \mathbf{t} * \mathbf{t} + tanh(\mathbf{t})$, where $\mathbf{t} = \mathbf{t}_s$ if the time transition is shorter than a preset threshold θ_t, otherwise $\mathbf{t} = \mathbf{t}_l$. Here, $*$ denotes the element-wise multiplication. Similarly, we obtain the representation for spatial intervals $\mathbf{z}^{\triangle d}$ with a spatial threshold θ_d. The distance between two POIs is formed by the Haversine distance [21]. After updating each user u's representation $\mathbf{h}_u^t$ and each POI l's representation $\mathbf{h}_l^t$, we recommend a list of POIs for user u to visit at the next timestamp $t + 1$ by selecting the POIs with the highest prediction scores computed from a fully connected layer. We model the prediction of the next POI as a multi-class classification problem, where the ground truth is the POI that each user actually visits at the next timestamp. We also adopt the cross-entropy loss for the model optimization [3].

4 EXPERIMENTS

We now evaluate the effectiveness of the proposed method in next POI recommendation by leveraging a real-world location data.

Table 3: Performance of next POI recommendation for different methods.

	Baltimore July		DC July		DC August	
	Recall@10	MRR@10	Recall@10	MRR@10	Recall@10	MRR@10
Markov	0.257 ± 0.001	0.088 ± 0.004	0.471 ± 0.012	0.224 ± 0.012	0.431 ± 0.013	0.224 ± 0.012
LightGCN	0.650 ± 0.044	0.392 ± 0.011	0.609 ± 0.021	0.291 ± 0.034	0.622 ± 0.019	0.311 ± 0.014
GRU4Rec	0.657 ± 0.009	0.384 ± 0.006	0.617 ± 0.019	0.300 ± 0.013	0.624 ± 0.005	0.306 ± 0.005
DeepMove	0.715 ± 0.031	0.402 ± 0.003	0.660 ± 0.007	0.333 ± 0.006	0.648 ± 0.014	0.331 ± 0.009
SASRec	0.738 ± 0.029	0.415 ± 0.003	0.694 ± 0.019	0.335 ± 0.006	0.687 ± 0.006	0.353 ± 0.007
TiSASRec	0.836 ± 0.002	0.420 ± 0.007	0.760 ± 0.011	0.353 ± 0.014	0.763 ± 0.006	0.379 ± 0.003
STAN	0.868 ± 0.003	0.426 ± 0.004	0.807 ± 0.005	0.360 ± 0.007	0.816 ± 0.007	0.426 ± 0.004
MEMO	**0.891±0.004**	**0.446±0.009**	**0.831±0.002**	**0.380±0.006**	**0.840±0.003**	**0.435±0.008**
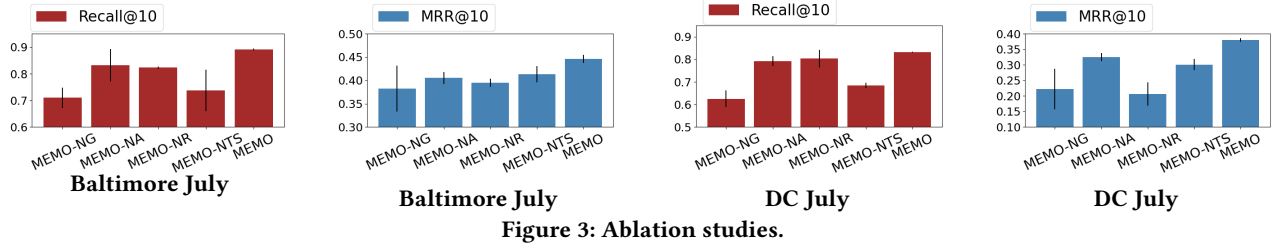
Figure 3: Ablation studies.
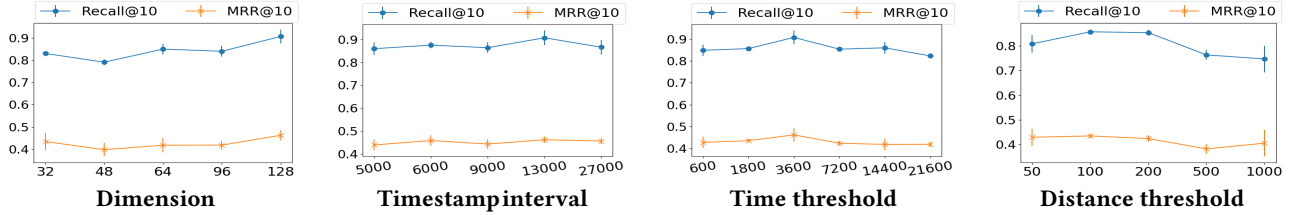
Figure 4: Parameter studies.

4.1 Experimental Setting

Data. The population-scale individual-level location data are provided by a leading data aggregator, who aggregates the data from more than 400 commonly used mobile apps, via a proprietary software development kit (SDK) installed on these apps to minimize battery drainage while tracking locations. The data are collected in compliance with privacy regulations, including General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA). The data cover one quarter of the U.S. population and are representative of the population. Each user is tracked on average every five minutes. Each data record contains an anonymized user ID common across all apps, timestamp, longitude, and latitude of the location visited, speed, and dwell time. Specifically, we analyze three datasets over different time periods in 2019: Baltimore (B'more) July, DC July, and DC August. Analyzing the users' trajectories via the state-of-the-art stop point detection method, Infostop [1], we detect the users' home and work locations as the most frequent stop points during the weekday nighttime and daytime respectively; and as a result infer their focal social relations, families and colleagues. Table 1 displays the summary statistics.

Baselines. We compare the proposed framework with the representative baselines across four categories (Table 2), including: 1) statistical methods, where the **Markov** [17] method recommends the next POIs by utilizing the dependency between every two consecutive visits; 2) Graph Network method, where **LightGCN** [6]

leverages the user-POI relation graph and a simplified GCN model with controllable oversmoothing for recommendations; 3) Recurrent Network methods, where **GRU4Rec** [7] and **DeepMove** [3] use RNNs to capture the users' preferences hidden in the trajectories; and 4) Self-attentive methods, including **SASRec** [9], **TiSASRec** [11], and **STAN** [15], which use self-attention to capture the long-term dependencies in the users' trajectories.

Parameter Settings. For each dataset, we partition the training/validation/test set into 80%/10%/10%; and set the training epoch as 50 and learning rate as 0.001. All reported results include the mean and standard deviation over 10 repeated runs. We use the Adam optimizer. By default, we set the representation dimension $d = 128$, length of each of the 400 timestamps as $13,000$ seconds, time threshold $\theta_t = 3600$ seconds, and distance threshold $\theta_d = 200$ meters. We use two popular metrics, Recall@K and MRR@K, to gauge the model performance, where K is set to 10.

4.2 Performance of Next POI Recommendation

Table 3 compares the performance of the different methods. The primary findings include: 1) GRU4Rec and DeepMove consistently outperform the Markov chain model, as the latter focuses merely on the consecutive visits, whereas the RNN-based methods capture more patterns hidden in the trajectories; 2) LightGCN performs worse than the RNN-based models, suggesting that only utilizing the user-POI relations and static embedding limit the POI recommendation

performance; 3) STAN exhibits a competitive performance, as it learns the user embeddings for personalized recommendations, and explicitly leverages the spatio-temporal information with the self-attention mechanism; and 4) MEMO consistently outperforms all baselines, primarily because i) MEMO captures the users' preferences more comprehensively by integrating the users' representations learned from the different relations; ii) MEMO models the user-POI mutual influence over time, with the learned representations better capturing the latent status of the users/POIs.

4.3 Ablation Studies

We further conduct ablation studies to investigate the contributions of the different model components. Specifically, we compare MEMO with its variants: (a) MEMO-NG in which the relation modeling component is removed; (b) MEMO-NA in which the self-attention mechanism is replaced with an ordinary attention mechanism based on a bilinear function; (c) MEMO-NR in which the mutual influence component is removed, i.e., the representations are learned just from the relation modeling component; and (d) MEMO-NTS in which the spatio-temporal representation is not used.

Given limited space, we only report the results on the Baltimore July and DC July data. The DC August data produce similar findings. Fig. 3 shows that MEMO-NG cannot achieve a satisfactory performance as it does not utilize the relational information. The performance of MEMO-NA also degrades as it implicitly assumes that all relation-specific representations are within the same latent space. The performance gap between MEMO and MEMO-NR indicates the importance of mutual influence modeling; and the comparison between MEMO-NTS and MEMO indicates the contribution of the spatio-temporal information to the recommendations.

4.4 Parameter Studies

To evaluate the robustness of our framework, we compare the performance under different hyperparameters. Fig. 4 shows the performance of MEMO with the varied representation dimension d in the range of $\{32, 48, 64, 96, 128\}$, the interval between two timestamps from $(5 \sim 27) \times 10^3$ seconds, the time threshold θ_t from 600 to 21600 seconds, and distance threshold θ_d from 50 to 1000 meters. We will illustrate the results with the Baltimore July data. The findings from the other two data sets remain similar. In general, the model is not very sensitive to the parameter settings; yet the performance can still benefit from a proper fine-tuning.

5 CONCLUSION

In this research, we focus on next POI recommendation by addressing two main challenges: modeling different types of relations and user-POI mutual influence over time. Methodologically, we propose a novel framework, MEMO, to address these challenges by leveraging a multi-relational representation learning module to model and aggregate different relations, and further enlisting the coupled RNNs to capture the inter-temporal user-POI mutual influence. The extensive experiments conducted on the population-scale, yet fine-grained individual-level location big data, supported by comprehensive model comparisons, ablation and parameter studies, validate the superiority of the proposed framework.

REFERENCES

[1] ASLAK, U., AND ALESSANDRETTI, L. Infostop: scalable stop-location detection in multi-user mobility data. *arXiv preprint arXiv:2003.14370* (2020).
[2] BAUMANN, P., KLEIMINGER, W., AND SANTINI, S. The influence of temporal and spatial features on the performance of next-place prediction algorithms. In *ACM UbiComp* (2013), pp. 449–458.
[3] FENG, J., LI, Y., ZHANG, C., SUN, F., MENG, F., GUO, A., AND JIN, D. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW* (2018), pp. 1459–1468.
[4] GAMBS, S., KILLIJIAN, M.-O., AND DEL PRADO CORTEZ, M. N. Next place prediction using mobility markov chains. In *MPM* (2012), pp. 1–6.
[5] HAMILTON, W. L., YING, R., AND LESKOVEC, J. Inductive representation learning on large graphs. In *NIPS* (2017), pp. 1025–1035.
[6] HE, X., DENG, K., WANG, X., LI, Y., ZHANG, Y., AND WANG, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval* (2020), pp. 639–648.
[7] HIDASI, B., KARATZOGLOU, A., BALTRUNAS, L., AND TIKK, D. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[8] HU, Z., DONG, Y., WANG, K., AND SUN, Y. Heterogeneous graph transformer. In *WWW* (2020), pp. 2704–2710.
[9] KANG, W.-C., AND MCAULEY, J. Self-attentive sequential recommendation. In *IEEE ICDM* (2018), pp. 197–206.
[10] KUMAR, S., ZHANG, X., AND LESKOVEC, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *ACM SIGKDD* (2019), pp. 1269–1278.
[11] LI, J., WANG, Y., AND MCAULEY, J. Time interval aware self-attention for sequential recommendation. In *WSDM* (2020), pp. 322–330.
[12] LI, R., SHEN, Y., AND ZHU, Y. Next point-of-interest recommendation with temporal and multi-level context attention. In *IEEE ICDM* (2018), pp. 1110–1115.
[13] LIAN, D., WU, Y., GE, Y., XIE, X., AND CHEN, E. Geography-aware sequential location recommendation. In *ACM SIGKDD* (2020), pp. 2009–2019.
[14] LIM, N., HOOI, B., NG, S.-K., WANG, X., GOH, Y. L., WENG, R., AND VARADARAJAN, J. Stp-udgat: Spatial-temporal-preference user dimensional graph attention network for next poi recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020), pp. 845–854.
[15] LUO, Y., LIU, Q., AND LIU, Z. Stan: Spatio-temporal attention network for next location recommendation. In *WWW* (2021), pp. 2177–2185.
[16] MASSA, P., AND AVESANI, P. Trust-aware recommender systems. In *ACM RecSys* (2007), pp. 17–24.
[17] MATHEW, W., RAPOSO, R., AND MARTINS, B. Predicting future locations with hidden markov models. In *ACM UbiComp* (2012), pp. 911–918.
[18] MCPHERSON, M., SMITH-LOVIN, L., AND COOK, J. M. Birds of a feather: Homophily in social networks. *Annual review of sociology 27*, 1 (2001), 415–444.
[19] SUN, K., QIAN, T., CHEN, T., LIANG, Y., NGUYEN, Q. V. H., AND YIN, H. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *AAAI* (2020), vol. 34, pp. 214–221.
[20] TANG, J., GAO, H., AND LIU, H. mtrust: Discerning multi-faceted trust in a connected world. In *ACM WSDM* (2012), pp. 93–102.
[21] VAN BRUMMELEN, G., AND HAMM, E. A. Heavenly mathematics: the forgotten art of spherical trigonometry. *Aestimatio: Critical Reviews in the History of Science 11* (2014), 127–130.
[22] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. In *NIPS* (2017), pp. 5998–6008.
[23] VELIČKOVIĆ, P., CUCURULL, G., CASANOVA, A., ROMERO, A., LIO, P., AND BENGIO, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[24] XIE, M., YIN, H., WANG, H., XU, F., CHEN, W., AND WANG, S. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (2016), pp. 15–24.
[25] YANG, C., BAI, L., ZHANG, C., YUAN, Q., AND HAN, J. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *ACM SIGKDD* (2017), pp. 1245–1254.
[26] YANG, Z., YANG, D., DYER, C., HE, X., SMOLA, A., AND HOVY, E. Hierarchical attention networks for document classification. In *NAACL-HLT* (2016), pp. 1480–1489.
[27] ZHAO, P., LUO, A., LIU, Y., ZHUANG, F., XU, J., LI, Z., SHENG, V. S., AND ZHOU, X. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE TKDE* (2020).
[28] ZHU, Y., LI, H., LIAO, Y., WANG, B., GUAN, Z., LIU, H., AND CAI, D. What to do next: Modeling user behaviors by time-lstm. In *IJCAI* (2017), vol. 17, pp. 3602–3608.