Memory-enhanced Period-aware Graph Neural Network for General POI Recommendation

Tianchi Yang[1], Haihan Gao[2], Cheng Yang[1], Chuan Shi[⋆1], Qianlong Xie[2], Xingxing Wang[2], Dong Wang[2]

[1]Beijing University of Posts and Telecommunications, China [2]Meituan, China
{yangtianchi, shichuan}@bupt.edu.cn albertyang33@gmail.com {gaohaihan, xieqianlong, wangxingxing04, wangdong07}@meituan.com,

Abstract. As the popularity of Location-based Services increases, Point-of-Interest (POI) recommendations receive higher requirements to characterize the users, POIs and interactions. Although many recent graph neural network-based (GNN-based) studies have tried working on temporal and spatial factors, they still cannot seamlessly handle the temporal locality and spatial consistency. To tackle this issue, we propose a novel Memory-enhanced Period-aware Graph neural network for general POI Recommendation (MPGRec). Specifically, it exploits the advantages of the GNN module in characterizing user preferences. Moreover, we develop a period-aware gate mechanism after the GNN information propagation to characterize the temporal locality, and devise a dynamic memory module to extract, store and disseminate global information for spatial consistency. Furthermore, we propose a reading and writing strategy to merge the GNN module and memory module into a unified framework. Extensive experiments are conducted on four real-world datasets, and the experimental results demonstrate the effectiveness of our method.

Keywords: Recommendation systems · Graph neural network

1 Introduction

With the popularity of Location-based Services, Point of interest (POI) recommendation has already drawn lots of research attention [3, 8, 5, 15].

There exist two research branches of POI recommendation. One focuses on sequential characteristics for temporal POI visit behavior mining, namely sequential POI recommendation [5, 2, 1, 11]. The other focuses on the general characteristics of the users, POIs and interactions under temporal and spatial factors. Some existing works also refer to it as general POI recommendations [8]. In this work, we study the general POI recommendation problem. Owing to the rapid development of graph neural networks (GNNs), GNN-based recommendation approaches have attracted the interest of many researchers, which have advanced

⋆ Corresponding author.

performance in capturing the general characteristics of interactions [4, 18, 7, 16, 14, 17]. Therefore, some recent studies [3, 8, 9] begin to explore GNN models for general POI recommendation, which introduce spatio-temporal contexts into the graph construction or message passing mechanism. In this way, GNNs can well characterize the user preferences under temporal and spatial factors.

Despite the great success of the existing methods, they still fail to consider the following two characteristics. (1) The interactions between users and POIs are not exactly the same among different time periods, e.g., morning and evening. We name this characteristic temporal locality. However, most GNN-based methods [3, 8] will aggregate neighbor information of all time periods and finally embed them into a single representation. It causes the information specific to different time periods to be confused, thus the temporal locality information also being ignored. (2) There is overlapping information among users and POIs that have not directly interacted or even are far away from each other, which can be treated as valuable and global information. We call this characteristic spatial consistency. However, existing GNN-based methods [3, 8] only pass messages for nodes on explicit interaction relationships (i.e., edges). It makes the nodes difficult to exploit valuable and global information from remote nodes, thus leading to difficulties in capturing spatial consistency.

Based on the above discussion, we propose to study the general POI recommendation based on the GNN framework to consider the temporal locality and spatial consistency. However, this is challenging due to the following questions. (1) *How to correctly distinguish information related to different time periods in the GNN, so as to consider temporal locality?* On the one hand, the GNN model cannot be trained on subgraphs divided by time period, since it separates the information related to different time periods into independent channels. On the other hand, it will still lead to a confusingly mixing of information to make time-specific transformations based on the whole graph like [3] for the neighbor information. Therefore, a special period-aware mechanism is needed that propagates the information of different periods, but outputs the information of a specific period for recommendation. (2) *How to make GNN not limited to the explicit interaction relationships in graph, thus further considering spatial consistency?* In the field of computer vision, the memory networks [12, 10] are recently explored to record global information across samples to improve few-shot learning tasks. Enlightened by it, we can transfer this idea to leverage a memory module to extract and store global node information that satisfies spatial consistency. Ultimately, it can serve as a springboard to propagate information to other nodes that are not explicitly connected. (3) *How to merge the GNN module and Memory module into a unified framework?* The existing memory approaches [6] are mainly implemented by a learnable parameter matrix, whose information is learned from the optimizer rather than the GNN. This results in an information gap between the memory module and the GNN module, which prevents them from becoming a unified framework. Therefore, we propose a dynamic memory module, which stores the valuable and global information learned from GNN in memory and then feeds back into user/POI representations.

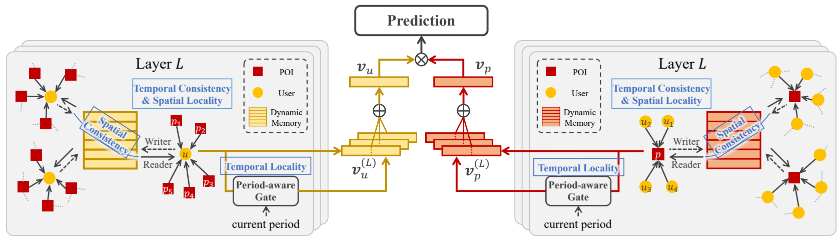

Fig. 1. Illustration of MPGRec.

To address the aforementioned challenges, we propose a novel Memory-enhanced Period-aware Graph neural network to solve the task of general POI Recommendation (MPGRec). Specifically, first of all, a user-POI interaction graph is built to depict the user interaction history. Then, a novel memory-enhanced period-aware graph neural network is proposed to learn the user and POI embeddings. To further characterize the temporal locality, we develop a period-aware gate mechanism after information propagation to separate the mixed information of different periods. Moreover, a dynamic memory module is devised to extract and store the valuable and global information from the GNN module and accordingly send it back to all nodes, which takes the spatial consistency into account. Finally, we propose a writing strategy and a reading strategy to merge the GNN module and memory module into a unified framework, where the writing strategy is to extract global information learned by the GNN and store it in memory, and the reading strategy is to send the information that is valuable to a particular node. The main contributions are summarized as follows:

- We propose a novel Memory-enhanced Period-aware Graph neural network for POI recommendation (MPGRec). It designs a period-aware gate mechanism for temporal locality and a dynamic memory module for spatial consistency.
- We propose a writing / reading strategy for the dynamic memory module, which merges GNN module and memory module into a unified framework.
- Experimental results on four benchmark datasets show that our MPGRec significantly outperforms state-of-the-art baselines and demonstrate the effectiveness of the proposed method.

2 Methodology

In this section, we present the proposed MPGRec. Specifically, as illustrated in Figure 1, based on a user-POI interaction graph, a novel memory-enhanced period-aware graph neural network is proposed to learn the user and POI embeddings. In detail, a period-aware gate mechanism is designed for the temporal locality to filter out information related to other periods after the message passing process in GNN. Meanwhile, a dynamic memory module is introduced to store and disseminate the valuable and global information learned by the GNN module for all nodes, which takes advantage of spatial consistency. Finally, in order to merge the GNN module and memory module into a unified framework, we propose a writing strategy based on the principle of maximizing the expressiveness of memory and a reading strategy based on the principle of information correlation, which enables the memory to extract global information learned by the GNN and also enables the GNN to successfully benefit from the memory.

2.1 Problem Formulation and Graph Construction

In this paper, we focus on the task of general POI recommendation [3], which aims to recommend POIs to a given user at a specific time period. Formally, let $\mathcal{U} = \{u_1, \ldots, u_{|\mathcal{U}|}\}$ denote a set of users, $\mathcal{P} = \{p_1, \ldots, p_{|\mathcal{P}|}\}$ denote a set of POIs, and $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ denote a set of time periods which are obtained by dividing the timestamps into a specific range. Given a user $u \in \mathcal{U}$ and current time period $t \in \mathcal{T}$, the problem is defined to recommend POIs $p \in \mathcal{P}$ that u would be interested at t. We are aware of that explicit consideration for user/POI location might improve the POI recommendation [3, 8], but we leave this extension in the future since it is not the key point focused in this work.

For modeling the user interaction history, similar to existing GNN-based recommendation methods [4], we construct a user-POI interaction graph, where each edge representing an interaction instance connects its corresponding user and POI. Formally, following [4], the user-POI interactions are modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V} = \mathcal{U} \bigcup \mathcal{P}$ represents the vertex set, where $\mathcal{U}$ is user set, and $\mathcal{P}$ is POI set. For each interaction instance that a user $u \in \mathcal{U}$ interacted a POI $p \in \mathcal{P}$, we build an edge $e = (u, p) \in \mathcal{E}$ to represent it. Note that following [4], we do not explicitly model the time period t into the graph.

2.2 Memory-enhanced Period-aware Graph Neural Network

Formally, the neighborhood aggregation and output rule of the l-th layer can be abstracted as follows,

$$\boldsymbol{v}_u^{(l+1)} = f_{\text{agg}}(\boldsymbol{v}_u^{(l)}, \{\boldsymbol{v}_p^{(l)} | p \in \mathcal{N}_u\}; \boldsymbol{M}_u^{(l)}), \quad \hat{\boldsymbol{v}}_u^{(l+1)} = g_{\text{out}}(\boldsymbol{v}_u^{(l+1)}; t), \tag{1}$$

$$\boldsymbol{v}_p^{(l+1)} = f_{\text{agg}}(\boldsymbol{v}_p^{(l)}, \{\boldsymbol{v}_u^{(l)} | u \in \mathcal{N}_p\}; \boldsymbol{M}_p^{(l)}), \quad \hat{\boldsymbol{v}}_p^{(l+1)} = g_{\text{out}}(\boldsymbol{v}_p^{(l+1)}; t), \tag{2}$$

where $\boldsymbol{v}_u$ and $\boldsymbol{v}_p$ are the aggregated embeddings of user u and POI p by aggregation function f_{agg} while $\hat{\boldsymbol{v}}_u$ and $\hat{\boldsymbol{v}}_p$ are the output embeddings by period-aware mechanism function g_{out}. $\mathcal{N}_u$ and $\mathcal{N}_p$ are the corresponding neighbors, and $\boldsymbol{M}_u^{(l)} \in \mathbb{R}^{K \times D}$ and $\boldsymbol{M}_p^{(l)} \in \mathbb{R}^{K \times D}$ are the memory matrices of K slots for user and POI vertices, respectively. D denotes the embedding size and $t \in \mathcal{T}$ denotes the current time period.

In the following paragraphs, Eq.(1) is taken as an example to illustrate the calculation procedure. Given a target node u and current time period t, suppose that the memory reader outputs a set of information denoted as $\hat{\boldsymbol{M}}_u^{(l)} \in \mathbb{R}^{K \times D}$ for target user u and the corresponding information weight vector $\boldsymbol{\alpha} \in \mathbb{R}^{K \times 1}$

after memory module calculations (which will discuss in detail in the next subsection). f_{agg} is defined as follows,

$$f_{\text{agg}}(\boldsymbol{v}_u^{(l)}, \{\boldsymbol{v}_p^{(l)} | p \in \mathcal{N}_u\}; \boldsymbol{M}_u^{(l)}) = \sum_{p \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_p|}} \boldsymbol{v}_p^{(l)} + \boldsymbol{\alpha}^\top \hat{\boldsymbol{M}}_u^{(l)}. \tag{3}$$

It is worth noting that in the above formula, the first term represents the neighbor information propagation of the GNN model. The second term represents the enhancement of the memory module.

Additionally, since the neighbors are of various time periods, the obtained representations are mixed up with different information specific to different time periods, thus causing other time-related information to become noise for the current time period t_u. Therefore, we apply a period-aware gating function g_{out} after the information propagation, which is calculated as follows,

$$\hat{\boldsymbol{v}}_u^{(l+1)} = g_{\text{out}}(\boldsymbol{v}_u^{(l+1)}; t) = \boldsymbol{v}_u^{(l)} \| (\boldsymbol{v}_u^{(l)} \odot \boldsymbol{g}_t) \in \mathbb{R}^{2D}, \tag{4}$$

where $\boldsymbol{g}_t$ is the trainable gating vector towards the current time period t in the period-aware gating function $g_{\text{out}}(\cdot;\cdot)$. Symbol $\|$ denotes the concatenation operation to retain both the time period-specific information for temporal locality and the general information for other characteristics. In this paper, we use different groups of gating vectors for user and POI nodes, but share in different model layers. Consequently, there are a total of $2|\mathcal{T}|$ gating vectors.

Inspired by LightGCN [4], we do not explicitly integrate the self-connection information during message passing, but instead, combine this by summing the embeddings from each layer to form the final representation. Formally,

$$\hat{\boldsymbol{v}}_u = \sum\nolimits_{l=0}^{L} w_l \hat{\boldsymbol{v}}_u^{(l)}, \quad \hat{\boldsymbol{v}}_i = \sum\nolimits_{l=0}^{L} w_l \hat{\boldsymbol{v}}_i^{(l)}, \tag{5}$$

where $\hat{\boldsymbol{v}}^{(0)} = \boldsymbol{v}^{(0)} || \boldsymbol{v}^{(0)}$, L denotes the number of model layers, $w_l \geqslant 0$ denotes the importance of the l-th layer embedding in constituting the final embedding. Following [4], we set w_l uniformly as $1/(L+1)$.

2.3 Dynamic Memory Module

In this subsection, we present the proposed dynamic memory network designed for spatial consistency. Specifically, the core of the reading strategy is to determine which information in memory is relevant to a target node and how important it should be. Traditional memory methods mostly choose attention module to read memory and assign corresponding importance. We argue that the non-negativity of attention weights limits the capabilities of memory. Therefore, we propose a reader with a correlation-based reading strategy. Formally, given the current information stored in a K-slot memory $\boldsymbol{M} \in \mathbb{R}^{K \times D}$, the target node representation $\boldsymbol{v} \in \mathbb{R}^D$, the output messages $\hat{\boldsymbol{M}}$ and corresponding correaltion weights $\boldsymbol{\alpha}$ in Eq. (3) are defined as follows,

$$\hat{\boldsymbol{M}} = \boldsymbol{M} \cdot \boldsymbol{W}_v^{(l)} \ , \quad \boldsymbol{\alpha} = \frac{1}{K} \text{hardtanh}((\boldsymbol{M}\boldsymbol{W}_k^{(l)}) \cdot (\boldsymbol{v}\boldsymbol{W}_q^{(l)})^\top / D), \tag{6}$$

where $\boldsymbol{W}_v^{(l)}$, $\boldsymbol{W}_k^{(l)}$, $\boldsymbol{W}_q^{(l)}$ is the parameter matrix for value, key, query, respectively. Different from the similarity-based readers that usually use the softmax function, we choose the hardtanh function to measure correlation, where positive weight $\boldsymbol{\alpha}$ represents positive correlation, while negative weight represents negative correlation. Thereby, the ability of memory enhancement increases since it relaxes the non-negative constraint on the weights.

As for writing strategy, the intuition is that more nodes can benefit from the memory only when information stored in memory should have the greatest possible coverage, which is named the principle of maximum memory expressiveness. Obviously, the K records with the maximum memory expressiveness should have the smallest sum of the pair-wise correlations with each other. To this end, the writing strategy finds a memory slot having the most overlapping information with others, and updates it with the least overlapping information. Formally, given the current information stored in the K-slot memory $\boldsymbol{M} \in \mathbb{R}^{K \times D}$ and the representation matrix $\boldsymbol{V} \in \mathbb{R}^{N \times D}$ of N nodes outputted by the GNN module,

$$\boldsymbol{m}^* = \arg\max_i \sum\nolimits_{j \neq i} |\mathrm{corr}(\boldsymbol{m}_i, \boldsymbol{m}_j)|, \tag{7}$$

where $\boldsymbol{m}_i \in \mathbb{R}^{1 \times D}$ denotes a slot of $\boldsymbol{M}$, and the correlation is measured by $\mathrm{corr}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$. Note that information with strong negative correlation is also worth storing, since our correlation-based reader allows negative weights. Finally, the information that is most worthy of being written in memory should have the least overlapping information with the existing slots, formally,

$$\boldsymbol{v}^* = \arg\min_i \sum\nolimits_{\boldsymbol{m}_j \neq \boldsymbol{m}^*} |\mathrm{corr}(\boldsymbol{v}_i, \boldsymbol{m}_j)|, \tag{8}$$

where $\boldsymbol{v}_i$ denotes a row of matrix $\boldsymbol{V}$. In order to ensure the updating convergence of memory, the above update operation will be performed if and only if

$$\sum\nolimits_{\boldsymbol{m}_j \neq \boldsymbol{m}^*} |\mathrm{corr}(\boldsymbol{m}^*, \boldsymbol{m}_j)| > \sum\nolimits_{\boldsymbol{m}_j \neq \boldsymbol{m}^*} |\mathrm{corr}(\boldsymbol{v}^*, \boldsymbol{m}_j)|. \tag{9}$$

Besides, for the consideration of efficiency and stability, the writer updates $1 \leqslant k \ll K$ slots once a step according to the above rules. Note that for initialization, we randomly select K node representations from GNN module to build the initial memory, and then it will be trained by the above writing strategy.

2.4 Model Training

After propagating L layers, the user embeddings $\hat{\boldsymbol{v}}_u$ and POI embeddings $\hat{\boldsymbol{v}}_p$ have been obtained. In this section, for model training, we employ the Bayesian Personalized Ranking (BPR) loss [13], which is a pairwise loss that encourages the prediction of an observed entry to be higher than its unobserved counterparts. Formally,

$$L_{BPR} = -\sum\nolimits_{u \in \mathcal{U}; p \in \mathcal{N}_u; q \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{up} - \hat{y}_{uq}), \text{ where } \hat{y}_{up} = \hat{\boldsymbol{v}}_u^\top \hat{\boldsymbol{v}}_p, \tag{10}$$

Adam optimizer is adopted to train the model in a mini-batch manner.

3 Experiments

3.1 Experimental Setup

Datasets To evaluate the effectiveness of MPGRec, we conduct experiments on four benchmark POI recommendation datasets: **Foursquare**, **Gowalla**, **Yelp** and **Meituan**[1]. For datasets Foursquare, Gowalla and Yelp, we use the datasets released by a public tool RecBole [19], and eliminate those users with less than 10 check-in POIs, as well as those POIs with less than 10 visitors. We process their check-in timestamps at equal time intervals to obtain 4 periods. For dataset Meituan, we use the raw dataset without elimination. The statistics after preprocessed are reported in Table 1.

Table 1. Statistics of datasets.

Dataset	# User	# POI	# Interaction	Density
Foursquare	830	1,090	21,367	2.4E-02
Gowalla	29,859	40,989	1,027,464	8.4E-04
Yelp	72,093	41,234	1,842,016	6.2E-04
Meituan	200,000	3,559	1,240,419	1.7E-03

Evaluation Metrics In the experiments, we randomly select 70%, 10% and 20% of interactions of each user for training, validation and testing, respectively. We choose the following widely used evaluation metrics: Recall@N, NDCG@N and HR@N for N=1, 3, 5, 10 and 20. Following [4], all items that are not interacted with by a user are the candidates and a prediction is correct only when the item and the corresponding time period are both correct.

Baselines We select the following baselines for POI recommendation to validate the effectiveness of our model: GNN-based general recommendation methods: LightGCN [4], SimGCL [18], HMLET [7]; GNN-based POI recommendation method: STGCN [3]; memory-based method: MMCF [6].

Implementation Detail Following existing settings [4] for fair comparison, we set the dimension of node embeddings $D = 64$, the number of layers $L = 2$ for all datasets for all methods. For our model, we set the number of memory slots $K = 50$ and update $k = 2$ slots once a training step. L2 regularization coefficient is set $1e^{-5}$. The learning rate is set $1e^{-3}$ for Foursquare, $5e^{-4}$ for Gowalla and Yelp, $1e^{-4}$ for Meituan. The batch size is set 2048.

3.2 Overall Results

Table 2 shows the recommendation performance on the four datasets. The observations and conclusions are discussed as follows.

In all cases, the proposed MPGRec outperforms all the baselines across four datasets on all metrics. Specifically, MPGRec has achieved an average improvement of 20.5%, 5.4%, 3.7% and 5.3% on the four datasets, respectively, which validates the effectiveness and robustness of our model. In detail, MPGRec has a larger improvement ratio when the N (for top N) is smaller, e.g., in terms of Recall@N on Meituan, the performance has more than double improvement when N=1 than N=3. We attribute it to the consideration of both temporal locality and spatial consistency, which benefits MPGRec to make more accurate predictions. Besides, the GNN-based methods, especially the powerful LightGCN,

[1] https://www.biendata.xyz/competition/smp2021_1/

Table 2. Overall recommendation performance on the four datasets. The best and second best results are bold and underlined, respectively. We also report improvement of MPGRec compared to the best baseline method.

Model	Recall@N					NDCG@N					HR@N				
	1	3	5	10	20	1	3	5	10	20	1	3	5	10	20
Dataset: Foursquare															
LightGCN	0.0214	0.0506	0.0700	0.0997	0.1563	0.0347	0.0456	0.0533	0.0645	0.0814	0.0347	0.0905	0.1222	0.1760	0.2634
SimGCL	0.0243	0.0495	0.0704	0.1153	0.1707	0.0372	0.0461	0.0548	0.0707	0.0874	0.0372	0.0860	0.1227	0.1926	0.2800
HMLET	0.0211	0.0475	0.0693	0.1158	0.1731	0.0362	0.0431	0.0522	0.0697	0.0866	0.0362	0.0804	0.1176	0.1946	0.2775
STGCN	0.0184	0.0421	0.0571	0.0966	0.1455	0.0327	0.0385	0.0440	0.0588	0.0736	0.0327	0.0719	0.0960	0.1664	0.2479
MMCF	0.0178	0.0513	0.0711	0.0997	0.1521	0.0341	0.0440	0.0526	0.0631	0.0773	0.0345	0.0877	0.1193	0.1697	0.2535
MPGRec	**0.0301**	**0.0613**	**0.0840**	**0.1378**	**0.1954**	**0.0493**	**0.0571**	**0.0662**	**0.0857**	**0.1030**	**0.0493**	**0.1071**	**0.1438**	**0.2278**	**0.3172**
Impr.	23.87%	19.49%	18.14%	19.00%	12.88%	32.53%	23.86%	20.80%	21.22%	17.85%	32.53%	18.34%	17.20%	17.06%	13.29%
Dataset: Gowalla															
LightGCN	0.0278	0.0555	0.0750	0.1087	0.1559	0.0592	0.0602	0.0663	0.0775	0.0916	0.0592	0.1159	0.1541	0.2178	0.2973
SimGCL	0.0243	0.0509	0.0697	0.1037	0.1490	0.0546	0.0558	0.0615	0.0727	0.0862	0.0546	0.1077	0.1431	0.2040	0.2762
HMLET	0.0258	0.0527	0.0712	0.1058	0.1535	0.0560	0.0573	0.0628	0.0745	0.0888	0.0560	0.1107	0.1474	0.2119	0.2906
STGCN	0.0178	0.0377	0.0514	0.0762	0.1101	0.0388	0.0408	0.0450	0.0532	0.0634	0.0388	0.0823	0.1114	0.1619	0.2270
MMCF	0.0031	0.0078	0.0115	0.0208	0.0366	0.0071	0.0084	0.0098	0.0131	0.0179	0.0071	0.0187	0.0279	0.0495	0.0852
MPGRec	**0.0290**	**0.0583**	**0.0787**	**0.1167**	**0.1691**	**0.0613**	**0.0631**	**0.0695**	**0.0821**	**0.0978**	**0.0613**	**0.1221**	**0.1620**	**0.2300**	**0.3146**
Impr.	4.32%	5.05%	4.93%	7.36%	8.47%	3.55%	4.82%	4.83%	5.94%	6.77%	3.55%	5.35%	5.13%	5.60%	5.82%
Dataset: Yelp															
LightGCN	0.0102	0.0232	0.0335	0.0532	0.0821	0.0196	0.0228	0.0268	0.0337	0.0426	0.0196	0.0446	0.0645	0.1018	0.1451
SimGCL	0.0105	0.0212	0.0284	0.0415	0.0590	0.0184	0.0205	0.0232	0.0278	0.0329	0.0184	0.0377	0.0505	0.0746	0.1060
HMLET	0.0117	0.0233	0.0314	0.0478	0.0719	0.0211	0.0232	0.0262	0.0320	0.0391	0.0211	0.0436	0.0599	0.0919	0.1364
STGCN	0.0018	0.0062	0.0096	0.0181	0.0353	0.0109	0.0111	0.0118	0.0144	0.0202	0.0109	0.0310	0.0445	0.0789	0.1353
MMCF	0.0103	0.0237	0.0337	0.0531	0.0751	0.0191	0.0234	0.0277	0.0332	0.0395	0.0190	0.0460	0.0648	0.1001	0.1441
MPGRec	**0.0121**	**0.0250**	**0.0349**	**0.0543**	**0.0827**	**0.0222**	**0.0248**	**0.0286**	**0.0355**	**0.0439**	**0.0222**	**0.0470**	**0.0662**	**0.1033**	**0.1546**
Impr.	3.42%	5.49%	3.56%	2.07%	0.73%	5.21%	5.98%	3.25%	5.34%	3.05%	5.21%	2.17%	2.16%	1.47%	6.55%
Dataset: Meituan															
LightGCN	0.2797	0.3346	0.3538	0.3770	0.4005	0.3039	0.3202	0.3284	0.3365	0.3428	0.3039	0.3605	0.3801	0.4034	0.4265
SimGCL	0.2699	0.2975	0.3065	0.3189	0.3345	0.2949	0.2934	0.2971	0.3014	0.3055	0.2949	0.3221	0.3316	0.3447	0.3613
HMLET	0.2813	0.3330	0.3519	0.3764	0.4014	0.3064	0.3198	0.3280	0.3364	0.3431	0.3064	0.3594	0.3786	0.4027	0.4273
STGCN	0.1840	0.2663	0.3002	0.3396	0.3737	0.1995	0.2377	0.2522	0.2657	0.2749	0.1995	0.2879	0.3237	0.3650	0.3997
MMCF	0.2760	0.3487	0.3629	0.3740	0.3843	0.3053	0.3293	0.3352	0.3391	0.3418	0.3053	0.3766	0.3899	0.4004	0.4103
MPGRec	**0.3056**	**0.3620**	**0.3766**	**0.3921**	**0.4117**	**0.3357**	**0.3492**	**0.3554**	**0.3608**	**0.3659**	**0.3357**	**0.3894**	**0.4033**	**0.4182**	**0.4374**
Impr.	8.64%	3.81%	3.78%	4.01%	2.57%	9.56%	6.04%	6.03%	6.40%	6.65%	9.56%	3.40%	3.44%	3.67%	2.36%

SIMGCL and HMLET, have generally achieved sub-optimal results. However, STGCN, which should be a SOTA method for POI recommendation, has never achieved the expected performance. We analyze that this is due to the terrible information mixing caused by its propagation rules, which in turn indicates the effectiveness of our designs such as the period-aware mechanism. MMCF, the SOTA memory-based method, has generally performed sub-optimally on Meituan but poorly on Gowalla. This polarized phenomenon, on the one hand, verifies the enhancement capability of the memory mechanism, and on the other hand, reflects the insufficiency of memory that is implemented by a learnable parameter matrix. This validates our design for the dynamic memory module.

3.3 Ablation Study

In this subsection, we compare our MPGRec with 4 variants to validate the characteristics we have summarized and investigate the effectiveness of each proposed module in our model. Due to the limited space of the paper, we only report the performance on metric HR@N since the performances on other metrics are consistent. Specifically, $MPGRec_{\setminus D}$ is a variant that replace the proposed dynamic memory module with the simple memory implemented as a trainable param-

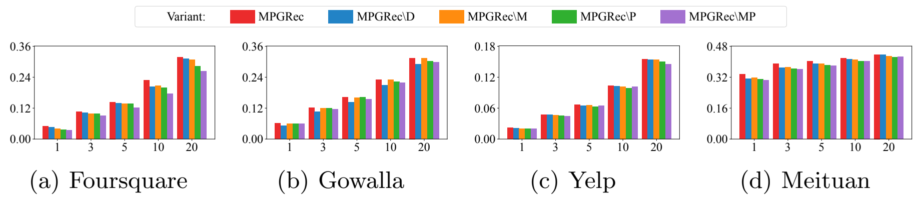

Fig. 2. Recommendation performance for ablation study.

eter matrix like [6]. $MPGRec_{\setminus M}$ directly removes the entire memory module, $MPGRec_{\setminus P}$ removes the period-aware gate mechanim only, and $MPGRec_{\setminus MP}$ removes both. As reported in Figure 2, the recommendation performance degrades with each module removed or modified, which verifies the effectiveness of each of the proposed modules. In detail, the performance of $MPGRec_{\setminus D}$ is sometimes even worse than $MPGRec_{\setminus M}$. This weird phenomenon is caused by the information gap between memory and GNN module since its expressiveness depends on the optimizer. It confirms the effectiveness of our reading and writing strategies in the dynamic memory module. Besides, the removal of either the dynamic memory module ($MPGRec_{\setminus M}$) or the period-aware mechanism ($MPGRec_{\setminus P}$) will reduce the performance, and removing both ($MPGRec_{\setminus MP}$) obtains the worst performance. This phenomenon on the one hand proves the necessity of the summarized characteristics; on the other hand, it also affirms the module effectiveness of each of our designs.

4 Conclusion

In this paper, we study the general POI recommendation based on the GNN framework and propose a novel Memory-enhanced Period-aware Graph neural network for general POI recommendation (MPGRec). In detail, it designs a period-aware gate mechanism for temporal locality, and a dynamic memory module for spatial consistency. Besides, we propose a correlation-based reading strategy and a writing strategy to maximize memory expressiveness, which merges the GNN module and memory module into a unified framework. Finally, we conduct extensive experiments to verify the effectiveness of our MPGRec.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, 62172052, 62002029, 62172052, U1936014) and also supported by Meituan.

References

1. Cui, Q., Zhang, C., Zhang, Y., Wang, J., Cai, M.: ST-PIL: Spatial-Temporal Periodic Interest Learning for Next Point-of-Interest Recommendation. In: CIKM. pp. 2960–2964 (Oct 2021)

2. Cui, Y., Sun, H., Zhao, Y., Yin, H., Zheng, K.: Sequential-Knowledge-Aware Next POI Recommendation: A Meta-Learning Approach. ACM Transactions on Information Systems **40**(2), 1–22 (Apr 2022)
3. Han, H., Zhang, M., Hou, M., Zhang, F., Wang, Z., Chen, E., Wang, H., Ma, J., Liu, Q.: STGCN: A Spatial-Temporal Aware Graph Learning Method for POI Recommendation. In: ICDM. pp. 1052–1057 (Nov 2020)
4. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In: SIGIR. pp. 639–648 (Jul 2020)
5. Islam, M.A., Mohammad, M.M., Sarathi Das, S.S., Ali, M.E.: A survey on deep learning based Point-of-Interest (POI) recommendations. Neurocomputing **472**, 306–325 (Feb 2022)
6. Jiang, X., Hu, B., Fang, Y., Shi, C.: Multiplex memory network for collaborative filtering. In: ICDM. pp. 91–99 (2020)
7. Kong, T., Kim, T., Jeon, J., Choi, J., Lee, Y.C., Park, N., Kim, S.W.: Linear, or Non-Linear, That is the Question! In: WSDM. pp. 517–525 (2022)
8. Li, Z., Cheng, W., Xiao, H., Yu, W., Chen, H., Wang, W.: You Are What and Where You Are: Graph Enhanced Attention Network for Explainable POI Recommendation. In: CIKM. pp. 3945–3954 (Oct 2021)
9. Lim, N., Hooi, B., Ng, S.K., Wang, X., Goh, Y.L., Weng, R., Varadarajan, J.: STP-UDGAT: Spatial-Temporal-Preference User Dimensional Graph Attention Network for Next POI Recommendation. In: CIKM. pp. 845–854 (Oct 2020)
10. Liu, X., Tian, X., Lin, S., Qu, Y., Ma, L., Yuan, W., Zhang, Z., Xie, Y.: Learn from Concepts: Towards the Purified Memory for Few-shot Learning. In: IJCAI. pp. 888–894 (Aug 2021)
11. Rahmani, H.A., Aliannejadi, M., Baratchi, M., Crestani, F.: A Systematic Analysis on the Impact of Contextual Information on Point-of-Interest Recommendation. ACM Transactions on Information Systems **40**(4), 1–35 (Oct 2022)
12. Ramalho, T., Garnelo, M.: Adaptive Posterior Learning: few-shot learning with a surprise-based memory module. In: ICLR (2019)
13. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian Personalized Ranking from Implicit Feedback. In: UAI. pp. 452–461 (2009)
14. Sun, L., Rao, Y., Zhang, X., Lan, Y., Yu, S.: MS-HGAT: Memory-enhanced Sequential Hypergraph Attention Network for Information Diffusion Prediction. In: AAAI (2022)
15. Sánchez, P., Bellogín, A.: Point-of-Interest Recommender Systems based on Location-Based Social Networks: A Survey from an Experimental Perspective. ACM Computing Surveys p. 3510409 (Jan 2022)
16. Wang, D., Wang, X., Xiang, Z., Yu, D., Deng, S., Xu, G.: Attentive sequential model based on graph neural network for next poi recommendation. In: WWW. vol. 24, pp. 2161–2184 (Nov 2021)
17. Wang, Z., Zhu, Y., Zhang, Q., Liu, H., Wang, C., Liu, T.: Graph-enhanced Spatial-temporal Network for Next POI Recommendation. ACM Transactions on Knowledge Discovery from Data pp. 1556–4681 (Feb 2022)
18. Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., Nguyen, Q.V.H.: Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In: SIGIR. pp. 1294–1303 (2022)
19. Zhao, W.X., Mu, S., Hou, Y., Lin, Z., Chen, Y., Pan, X., Li, K., Lu, Y., Wang, H., Tian, C., others: Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In: CIKM (2021)