HoST: Exploiting Heterogeneous Spatial-Temporal Graph for Next POI Recommendation

Anonymous Author(s)

ABSTRACT

Next Point-Of-Interest (POI) recommendation is a fundamental problem for location-based services, which aims to recommend the next POIs for users to visit based on their historical check-in sequences. Most existing methods mainly focus on how to make recommendations based on users' own historical check-in sequences, and have developed various sequential models. Recently, several studies construct graphs (e.g., POI-POI transition graph) from the check-in data and use them to augment sequential models, which have brought significant improvements. However, there are still several under-explored challenges. *(C1) How to construct fine-grained spatial-temporal context graphs?* Most existing methods construct large global graphs as the input to the model, which is neither efficient nor necessary for a single user. *(C2) How to build a unified spatial-temporal model?* Existing approaches follow the Sequence Augmented by Graph (SAG) paradigm, which uses sequential models as backbones and separate graph models as supplements. Their overall frameworks are usually complex. *(C3) How to model multi-granular temporal periodicity?* Users naturally have multi-granular movement patterns, i.e. daily and weekly patterns.

To address these challenges, we introduce a novel heterogeneous spatial temporal method called HoST. For *(C1)*, we introduce HoST-Graph to first construct a fine-grained global Heterogeneous Spatial Temporal Graph (HSTG) by exploiting all check-in sequences, and then sample a relative spatial-temporal context (i.e., ego-graph) for the target user and time. For *(C2)*, we propose to directly use the sampled ego-graph as the context to extract user embeddings regardless of the distant past check-ins. We refer to this new paradigm as Spatial-Temporal Context (STC) paradigm. Under the STC paradigm, we introduce a simple unified model HoST-GNN$_{\text{base}}$ based on Graph Attention Network (GAT), which is equipped with trainable edge type embeddings and spatial-temporal slot embeddings. To further address *(C3)*, multi-granular temporal slot embeddings are introduced and thus we have our full model HoST-GNN. Comprehensive experiments on benchmark datasets demonstrate the effectiveness of HoST.

ACM Reference Format:

Anonymous Author(s). 2018. HoST: Exploiting Heterogeneous Spatial-Temporal Graph for Next POI Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the widespread applications of positioning technology and mobile internet, Location-Based Services (LBS) have become an indispensable part of people's daily lives. LBS providers, such as Google Map, Yelp and Foursquare, enable users to share information on points of interests (POIs) and record their movement trajectories. To improve users' experience and relevant services such as advertising strategies [13], the next Point-Of-Interest (POI) recommendation systems have been developed to model users' mobility patterns and recommend them attractive POIs based on their historical check-in sequences and current spatial-temporal context.

Most existing approaches make recommendations based on users' own historical check-in sequences. Earlier works use Markov chains to capture the dependencies between observed visits and next POI decisions [2, 5]. Later, Recurrent Neural Network (RNN) based methods, which extends classic RNNs to percept spatial-temporal context [4, 17, 35, 50], become mainstream due to their strong capability of handling sequential data. Recently, several spatial-temporal-aware self-attention based and Transformer based methods are proposed to model both successive and non-successive transition patterns within individual check-in sequences [22, 26, 29, 44]. However, most users only have a few check-in records in real-world datasets. The highly sparse user-POI interaction data cannot provide sufficient supervision signals, and thus it is difficult to make satisfying recommendations solely based on users' own check-in sequences. To tackle the data sparsity issue, several graph-augmented approaches have been proposed recently, which leverage global user-POI collaborative information to learn general user mobility patterns [20, 23, 24, 30, 46]. Specifically, they construct global graphs (e.g., user-POI and POI-POI graphs) based on all users' check-in sequences such that relevant users' records could complement each other.

Despite significant improvements brought by graphs [9, 23, 24, 30, 46], there are still several under-explored challenges. *(C1) How to construct fine-grained spatial-temporal context graphs?* Most existing studies construct large coarse-grained global graphs [9, 20, 24, 46], such as the POI-POI transition matrix, which will inevitably lead to information loss. Although [30] builds a fine-grained knowledge graph, it uses various external tools [1, 43] to learn graph structure and uses the global graph as input to the model. When recommending the next POI for a single user, it is unnecessary and inefficient to use the entire global graph as the model's input. Therefore, a relative spatial-temporal context graph for the target user is desired. *(C2) How to build a unified spatial-temporal model?* The aforementioned methods follow a Sequence Augmented by Graph (SAG) paradigm, which means they mainly focus on modeling check-in sequences and treat graphs as auxiliary information. They usually have separate sequential and graph models, and the entire frameworks are usually overly complex. A simple unified model is always desired to simultaneously model the spatial and temporal dynamics and global collaborative information. *(C3) How*

Figure 1: A check-in sequence sample. The black and gray arrows represent transitions within short and long intervals. It shows the daily and weekly movement patterns.

to model multi-granular temporal periodicity? Human behavior patterns naturally exist in multiple temporal granularities. Figure 1 is an example check-in sequence of a user, which shows both daily and weekly patterns. At the daily granularity, the user goes to entertainment POIs during nights. At the weekly granularity, the user usually eats healthy/junk food for weekday/weekend lunches. Information from different granularities can be complementary, which is largely ignored by existing methods.

In this paper, we propose a novel heterogeneous spatial-temporal method, called HoST, to address these challenges. For *(C1)*, we introduce a graph construction method (HoST-Graph). We first build a fine-grained Heterogeneous Spatial-Temporal Graph (HSTG) to extract all the details of check-in sequences, including node types, edge (or interaction/transition) types, timestamps, geological locations, message propagation directions etc. Unlike [30], we do not leverage external tools to extract additional information. Rather than directly using the entire HSTG, we further sample a relative spatial-temporal context (i.e., ego-graph) to capture the relative spatial and temporal information for the target user at a target time. For *(C2)*, instead of following the SAG paradigm, we propose to directly build models and make recommendations based on the ego-graphs regardless of the distant past check-in records. We refer to this new paradigm as the Spatial-Temporal Context (STC) paradigm. Then we introduce a Graph Attention Network (GAT) based model HoST-GNN$_{base}$ to perform message propagation on and extract user embeddings from ego-graphs. HoST-GNN$_{base}$ is equipped with trainable edge type embeddings and spatial-temporal slot embeddings to encode the edge information and the relative spatial-temporal positional information, where each slot corresponds to a relative spatial or temporal interval. To further address *(C3)*, we introduce multi-granular temporal slot embeddings, where each temporal granularity is associated with a set of temporal slots. Then we have our full model HoST-GNN. Finally, we comprehensively evaluate HoST on two benchmark datasets to demonstrate its effectiveness.

In summary, our contribution are highlighted as follows:

- We introduce HoST-Graph to construct fine-grained heterogeneous spatial-temporal graphs. The full graph exploits details of check-in sequences and the sampled ego-graphs retain the relative spatial-temporal context for the target user and the target time.
- We propose a simple unified model HoST-GNN to extract embeddings from ego-graphs and make recommendations for the target user and time. HoST-GNN conduct inference only based on the sampled spatial-temporal context but not users' long check-in sequences.
- We conduct extensive experiments on two real-world benchmark datasets, and the results empirically demonstrate the superiority of our proposed HoST framework.

2 PRELIMINARIES

2.1 Problem Statement

Denote $\mathcal{U}$ and $\mathcal{L}$ as sets of users u and POIs l, and l is assigned with a spatial coordinate (latitude, longitude): $s_l = (lat_l, lon_l)$.

Definition 1 (Check-in). *A check-in is a tuple $c = (u, l, t)$, indicating that user u visits POI l at timestamp t.*

Definition 2 (Check-in Sequence). *A check-in sequence is a set of check-in records $c = (u, l, t)$ of a user $u \in \mathcal{U}$, which is arranged in chronological order. We denote $C_u = \{c_1, \cdots, c_{n_u}\}$ as the historical check-in sequence of u, and $C = \{C_u\}_{u \in \mathcal{U}}$ as the set of check-in sequences of all users. Here, n_u is the number of records of u.*

Definition 3 (Heterogeneous Spatial-Temporal Graph). *We define the Heterogeneous Spatial-Temporal Graph (HSTG) as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \phi_{\mathcal{V}}, \phi_{\mathcal{E}}, \xi_{\mathcal{V}}, \xi_{\mathcal{E}})$, where $\mathcal{V}$ and $\mathcal{E}$ are node set and directed edge set respectively. Each node and edge are assigned with a type $\phi_{\mathcal{V}}(v)$ and $\phi_{\mathcal{E}}(e)$. Available node and edge type sets are denoted as $\mathcal{P}_{\mathcal{V}} = \{\phi_{\mathcal{V}}(v), \forall v \in \mathcal{V}\}$ and $\mathcal{P}_{\mathcal{E}} = \{\phi_{\mathcal{E}}(e), \forall e \in \mathcal{E}\}$. Each node has attributes $\xi_{\mathcal{V}}(v)$ (e.g., spatial coordinates). Each directed edge $e_{vv'}$ is defined as a triplet $(v, v', \xi_{\mathcal{E}}(e))$, where v is the source node, v' is the target node, and $\xi_{\mathcal{E}}(e)$ is the edge attribute (e.g., timestamp t).*

Problem 1 (Next POI recommendation). *Given all the available historical check-in sequences C, the target user u, the corresponding check-in sequence C_u and a future timestamp T, the next POI recommendation task aims to recommend a ranked list of POIs and predict the most likely POI that user u would like to visit at T.*

2.2 Graph Attention Network

Graph Attention Network (GAT) [40] is one of the most popular GNN architectures. GAT is essentially a masked self-attention process, which updates node embeddings via the weighted-average over its neighbors' embeddings. Formally, let node embedding dimension be d, given node v and its neighbors $\mathcal{N}_v$, the attention score in the l-th single-head GAT layer is given by:

$$\alpha_{vv'} = \underset{v' \in \mathcal{N}_v}{\text{softmax}}(\sigma(\mathbf{a}^{(l)}[\mathbf{W}^{(l)}\mathbf{h}_v^{(l-1)} \parallel \mathbf{W}^{(l)}\mathbf{h}_{v'}^{(l-1)}])) \quad (1)$$

where $\mathbf{a}^{(l)} \in \mathbb{R}^{1\times 2d}, \mathbf{W} \in \mathbb{R}^{d\times d}$ are parameters, $\mathbf{h}_v^{(l-1)}, \mathbf{h}_{v'}^{(l-1)}$ are input node embeddings for the l-th layer, σ is the LeakyReLU activation function, and $\parallel$ is the concatenation operator. Given the learned scores, the neighborhood aggregation is conducted via a weighted sum of its neighboring node embeddings:

$$\mathbf{h}_v^{(l)} = \sum_{v' \in N_v} \alpha_{vv'}\mathbf{W}^{(l)}\mathbf{h}_{v'}^{(l-1)} \quad (2)$$

The single-head GAT layer can be easily extended to multi-head form by assigning separate parameters for attention vector $\mathbf{a}$ in each attention head. Given the head number N and corresponding learned scores $\alpha_{vv'}^n, n \in [1, N]$, the final output is a concatenation of the outputs of the N heads:

$$\mathbf{h}_v^{(l)} = \|_{n=1}^{N} \sum_{v' \in N_v} \alpha_{vv'}^n \mathbf{W}^{(l)}\mathbf{h}_{v'}^{(l-1)} \quad (3)$$

3 METHODOLOGY

In this section, we present our proposed HoST framework, which is comprised of a graph construction method HoST-Graph and a graph neural network model HoST-GNN. An illustration of HoST is presented in Figure 2. In section 3.1, we introduce HoST-Graph to construct fine-grained spatial-temporal context for *(C1)*. In section 3.2, we introduce a unified spatial-temporal graph neural network HoST-GNN, which could effectively capture the spatial-temporal context and multi-granular temporal patterns, to address *(C2)* and *(C3)*. In section 3.3, we show the objective function of our method.

3.1 Heterogeneous Spatial-Temporal Graph Construction

Most existing approaches construct coarse-grained global graphs [9, 23, 24, 30, 46] and a few [30] constructs fine-grained global graphs. They usually take the global graphs as inputs to models, which is neither necessary nor efficient for a single target user. By exploiting the check-in sequences, we first construct a global Heterogeneous Spatial-Temporal Graph (HSTG) $\mathcal{G}$, which is comprised of two basic graphs: interaction graph $\mathcal{G}_{inter}$ and transition graph $\mathcal{G}_{trans}$. For a target user and time, we sample an ego-graph $\mathcal{G}_{ego}$ from the global graph $\mathcal{G}$ to capture the relative spatial-temporal context. An illustration is shown in the upper part of Figure 2.

Interaction Graph. The user-POI directed interaction graph $\mathcal{G}_{inter}$ captures the interactions between users and locations, which also reflects user co-occurrences via the second-order relations. The set of nodes is the union of users $\mathcal{U}$ and locations $\mathcal{L}$: $\mathcal{V}_{inter} = \mathcal{U} \cup \mathcal{L}$. For each check-in record $c = (u, l, t)$, two directed edges (u, l, t) and (l, u, t) are constructed in $\mathcal{E}_{inter}$. To differentiate the two directions of information flow between the two kinds of nodes, we define two types of edge set $\mathcal{E}^{ul}_{inter} = \{(u, l, t) \in \mathcal{E}_{inter}\}$ and $\mathcal{E}^{lu}_{inter} = \{(l, u, t) \in \mathcal{E}_{inter}\}$, where $u \in \mathcal{U}, l \in \mathcal{L}$.

Transition Graph. The transition graph $\mathcal{G}_{trans}$ reflects global transition patterns among POIs. Its node set is $\mathcal{V}_{trans} = \mathcal{L}$. For each pair of POIs (l, l'), if l and l' appear successively in a check-in sequence $\exists C_u \subset C$, a forward edge $e_f = (l, l', t)$ and a backward edge $e_b = (l', l, t')$ are created in $\mathcal{E}_{trans}$, where t, t' are timestamps when the user creates the check-in records at l and l'. The forward and backward edges form two different edge sets $\mathcal{E}^{f}_{trans}$ and $\mathcal{E}^{b}_{trans}$.

Global HSTG. With the two basic graphs, node set and edge set of the global HSTG $\mathcal{G}$ are $\mathcal{V} = \mathcal{V}_{inter} \cup \mathcal{V}_{trans}$ and $\mathcal{E} = \mathcal{E}^{ul}_{inter} \cup \mathcal{E}^{lu}_{inter} \cup \mathcal{E}^{f}_{trans} \cup \mathcal{E}^{b}_{trans}$. There are totally two types of nodes and four types of edges in the global HSTG $\mathcal{G}$. We define our HSTG at time T as $\mathcal{G}(T) \subset \mathcal{G}$, where $\mathcal{V}(T) = \mathcal{V}$, $\mathcal{E}(T) = \{(v, v', t) \in \mathcal{E}, t < T\}$. We further define the edge type function $\phi_{\mathcal{E}}$, node type function $\phi_{\mathcal{V}}$, node attribute function $\xi_{\mathcal{V}}$ and edge attribute function $\xi_{\mathcal{E}}$ as:

$$\phi_{\mathcal{E}}(e_{ul}) = 0, \quad \phi_{\mathcal{E}}(e_{lu}) = 1, \quad \phi_{\mathcal{E}}(e_f) = 2, \quad \phi_{\mathcal{E}}(e_b) = 3 \tag{4}$$

$$\phi_{\mathcal{V}}(u) = 0, \quad \phi_{\mathcal{V}}(l) = 1 \tag{5}$$

$$\xi_{\mathcal{V}}(u) = \emptyset, \quad \xi_{\mathcal{V}}(l) = (lat_l, lon_l), \quad \xi_{\mathcal{E}}(e) = t \tag{6}$$

where $u \in \mathcal{U}, l \in \mathcal{L}, e_{ul} \in \mathcal{E}^{ul}_{inter}, e_{lu} \in \mathcal{E}^{lu}_{inter}, e_f \in \mathcal{E}^{f}_{trans}, e_b \in \mathcal{E}^{b}_{trans}$. Note that we allow multiple edges to exist simultaneously between any node pair in $\mathcal{G}$ to keep the fine-grained interaction information. By unifying various heterogeneous spatial-temporal information from $\mathcal{G}_{inter}$ and $\mathcal{G}_{trans}$ into a global HSTG $\mathcal{G}$, we could effectively capture the complex spatial-temporal context of each node beyond the local check-in history. Given a user u, we can easily find users who have visited the same POIs as u in the past, and further find their previous and next POI choice, which will be an important reference for the next POI prediction.

Ego-Graph. The amount of check-in sequences is enormous and thus the constructed HSTG $\mathcal{G}$ is huge. Processing the entire $\mathcal{G}$ for each user is extremely time-consuming. Besides, temporally recent check-in records have more impact on users' decisions than earlier records, and using all the historical records to model user preference may incur too much noise. Additionally, relative spatial-temporal information (e.g. time interval and physical distance) can reflect relative relations between global context information and current user state, which is more meaningful than absolute time and geographical coordinates. To address these issues, we propose to sample a relative spatial-temporal context (i.e. ego-graph $\mathcal{G}_{ego}$) for the target user from the global HSTG $\mathcal{G}$ and use the sampled ego-graph for training and inference. $\mathcal{G}_{ego}$ is built from the perspective of message propagation, where neighbors are sampled based on inward edges, and the edges function as message decorators to record relative temporal and spatial information.

Given the target user u with historical check-in sequence C_u and the future target time T, we first construct HSTG $\mathcal{G}(T)$. Then we treat u and T as the center node and center time of the ego-graph. Next, we sample a K-hop ego-graph $\mathcal{G}_{ego}(u, T)$ for the center u, T, where u forms the 0-th hop node set $\mathcal{N}^{(0)}_{ego} = \{u\}$. We iteratively sample the k-th hop neighbors $\mathcal{N}^{(k)}_{ego}$ based on the $k-1$-th hop neighbors $v \in \mathcal{N}^{(k-1)}_{ego}$. Considering that nodes receive messages from neighbors via inward edges under message propagation frameworks, for each node $v \in \mathcal{N}^{(k-1)}_{ego}$, we sample M temporally nearest inward edges $\mathcal{E}^{(k)}_{ego}(v) = \{\mathcal{E}^{(k)}_{ego-p}(v)\}_{p \in \mathcal{P}_{\mathcal{E}}}$ from $\mathcal{G}(T)$ for each edge type p, where $|\mathcal{E}^{(k)}_{ego-p}(v)| = M$. Specifically, if $v \in \mathcal{U}$, we sample the most recent edges from $\mathcal{E}^{lu}_{inter}(T)$; if $v \in \mathcal{L}$, then we sample from $\mathcal{E}_{trans}(T)$ and $\mathcal{E}^{ul}_{inter}(T)$. We denote the sampled neighbors as $\mathcal{N}^{(k)}_{ego}(v) = \{\mathcal{N}^{(k)}_{ego-p}(v)\}_{p \in \mathcal{P}_{\mathcal{E}}}$. The k-th hop node and edge sets are $\mathcal{N}^{(k)}_{ego} = \{\mathcal{N}^{(k)}_{ego}(v)\}_v$, $\mathcal{E}^{(k)}_{ego} = \{\mathcal{E}^{(k)}_{ego}(v)\}_v$, where $v \in \mathcal{N}^{(k-1)}_{ego}$.

Given $\mathcal{G}_{ego}(u, T)$, we further transform the absolute time and spatial coordinates into relative time and spatial distances, and use this additional information to decorate edges. For the time, we replace t of each edge $e = (v, v', t)$, with $\Delta t = T - t$. For the spatial coordinates, we first treat the spatial coordinates $S = (lat, lon)$ of u's last available check-in location as the center coordinates. Then, for each POI l, we compute its haversine distance Δs with S based on its coordinates $s_l = (lat_l, lon_l)$. Then Δs is put on edges originating from l (i.e., $l \to l'$ or $l \to u$) to decorate information originated from l. Finally, we have $\mathcal{G}_{ego}(u, T) = (\mathcal{V}, \mathcal{E}, \phi_{\mathcal{V}}, \phi_{\mathcal{E}}, \xi_{\mathcal{V}}, \xi_{\mathcal{E}})$, where $\mathcal{V} = \{\mathcal{N}^{(k)}_{ego}\}^{K}_{k=0}$, $\mathcal{E} = \{\mathcal{E}^{(k)}_{ego}\}^{K}_{k=0}$, type functions $\phi_{\mathcal{V}}, \phi_{\mathcal{E}}$ and node attribute function $\xi_{\mathcal{V}}$ are inherited from $\mathcal{G}(T)$. The edge attribute function $\xi_{\mathcal{E}}$ is re-defined as:

$$\xi_{\mathcal{E}}(e_{l*}) = (\Delta t, \Delta s), \quad \xi_{\mathcal{E}}(e_{ul}) = \Delta t \tag{7}$$

where u, l are user and location nodes, e_{l*} denotes the edge originating from l, e_{ul} is the $u \to l$ edge.

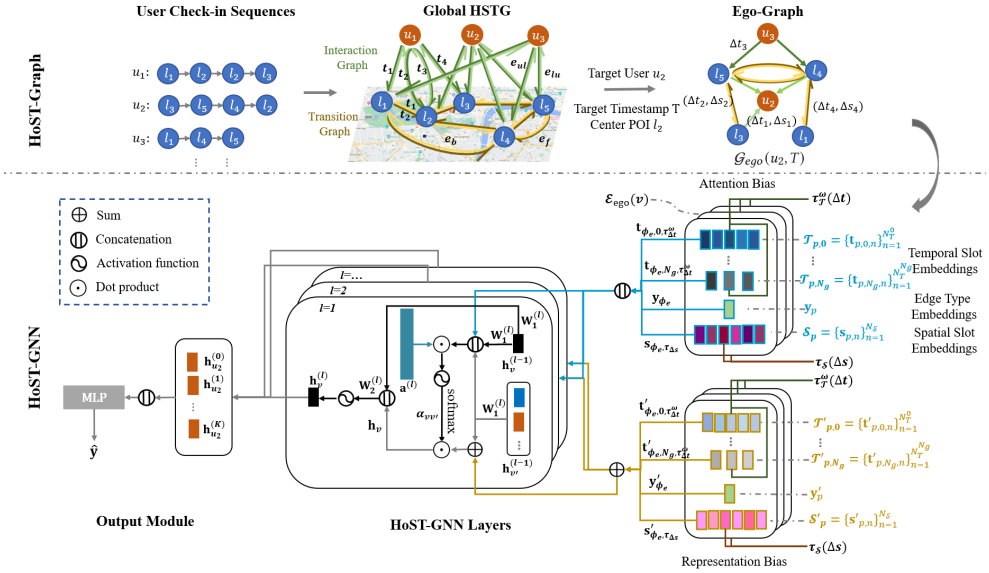


Figure 2: Overview of the HoST framework.

3.2 Heterogeneous Spatial Temporal Graph Neural Network

Following the Sequence Augmented by Graph (SAG) paradigm, most existing methods use sequential models as backbones to model users' long historical check-in sequences and separate graph models as supplements to model the relations among nodes (e.g., POI-POI, user-POI). These methods are often overly complex. In this paper, we propose to directly build a unified graph neural network for the ego-graph $\mathcal{G}_{ego}$ without intentionally using users' historical check-in sequences. We refer to this new paradigm as the Spatial-Temporal Context (STC) paradigm. To comprehensively capture the heterogeneous spatial-temporal information of $\mathcal{G}_{ego}$, a complex model is usually desired. However, as suggested by [27], with proper modification, the simple GAT could outperform those complex methods. Therefore, we first propose a simple yet effective base model (HoST-GNN$_{\text{base}}$) by introducing trainable edge type embeddings and spatial-temporal slot embeddings to the vanilla GAT. We further take into consider the multiple granularity of the temporal context, and introduce the full model HoST-GNN. The proposed HoST-GNN effectively addresses *(C2)(C3)*. An illustration of HoST-GNN is shown in the lower part of Figure 2.

HoST-GNN$_{base}$ Layer. The vanilla GAT layer does not distinguish edge types and thus fails to encode the semantics of edges (e.g., the $u \rightarrow l$ edge means"visit" and the $l \rightarrow l'$ edge means "transition"). To address this issue, we allocate an attention bias embedding $\mathbf{y}_p \in \mathbb{R}^d$ and a representation bias embedding $\mathbf{y}'_p \in \mathbb{R}^d$ for each edge type $p \in \mathcal{P}_{\mathcal{E}}$ to introduce edge type information into attention coefficients and node embeddings respectively. These embeddings are trainable and shared in each graph attention layer. Given an edge e, its corresponding type embeddings are thus $\mathbf{y}_{\phi_e}$ and $\mathbf{y}'_{\phi_e}$.[1] Note that we don't pay specific attention to the node types since they are encoded in the edge types (see Equation (4)-(5)).

To encode the spatial-temporal information flow, we introduce sets of trainable spatial slot embeddings and temporal slot embeddings to decorate the vanilla node embedding messages propagated from neighbors based on their own spatial-temporal knowledge. We only show how to obtain the temporal slot emebddings, and the spatial slot embeddings are obtained in a similar way. Given an interval ∂t and the maximum number of slots $N_{\mathcal{T}}$, we initialize two sets of slot embeddings $\mathcal{T} = \{\mathbf{t}_n\}_{n=1}^{N_{\mathcal{T}}}$, $\mathcal{T}' = \{\mathbf{t}'_n\}_{n=1}^{N_{\mathcal{T}}}$, where $\mathbf{t}_n, \mathbf{t}'_n \in \mathbb{R}^d$ are trainbale attention and representation bias embeddings associated with the n-th slot, and the time range of each slot is ∂t. $\mathcal{T}$ and $\mathcal{T}'$ have a similar role as $\mathbf{y}_p$ and $\mathbf{y}'_p$, which can also be regarded as relative positional embeddings [33] . However, positional information alone is not able to capture the semantics of information flow. For a fixed location l, the most impact edge type varies for different temporal slots. For example, for an amusement park, between 8:00-15:00, the $u \rightarrow l$ edge is dominating as people are entering the park, but after 16:00, $l \rightarrow l'$ is more influential since people are leaving the park and heading towards other locations. Therefore, it is necessary to further split slot embeddings for each edge type. For each edge type $p \in \mathcal{P}_{\mathcal{E}}$, we have edge type specific slot embeddings $\mathcal{T}_p = \{\mathbf{t}_{p,n}\}_{n=1}^{N_{\mathcal{T}}}$, $\mathcal{T}'_p = \{\mathbf{t}'_{p,n}\}_{n=1}^{N_{\mathcal{T}}}$. The slot index for the relative time Δt of an edge $e = (v, v', \Delta t, \Delta s)$ from the ego-graph $\mathcal{G}_{ego}$ is given by the temporal slot mapping function:

$$\tau_{\mathcal{T}}(\Delta t) = \min(\lceil \Delta t / \partial t \rceil, N_{\mathcal{T}}) \tag{8}$$

where $\lceil\ \rceil$ denotes round-up operation. We denote the temporal slot embeddings associated with e as $\mathbf{t}_{\phi_e,\tau_{\Delta t}}$, $\mathbf{t}_{\phi_e,\tau_{\Delta t}}$.[2]

Similar to the temporal slot embeddings, we can also obtain the edge type specific spatial slot embeddings for each edge type p:

[1] For clarity, we simplify the sub-script $\phi_{\mathcal{E}}(e)$ as ϕ_e in the following text.
[2] For clarity, we denote $\tau_{\Delta t}$ and $\tau_{\Delta s}$ as the temporal and spatial slot index respectively.

$\mathcal{S}_p = \{\mathbf{s}_{p,n}\}_{n=1}^{N_S}, \mathcal{S}'_p = \{\mathbf{s}'_{p,n}\}_{n=1}^{N_S}$, where N_S is the number of spatial slots, and the distance range of each slot is ∂s. We also have the spatial slot mapping function $\tau_{\mathcal{S}}$. Note that edges originating from u (i.e., e_{ul}) do not have the Δs attribute (Equation (7)), we use a zero vector $\mathbf{s}_0 = \mathbf{0}$ as their spatial slot embedding.

Given a node v and its inward edge $e = (v', v, \Delta t, \Delta s)$ from $\mathcal{G}_{ego}$, we could obtain the edge type embeddings $\mathbf{y}_{\phi_e}, \mathbf{y}'_{\phi_e}$, the spatial slot embeddings $\mathbf{s}_{\phi_e,\tau_{\Delta s}}, \mathbf{s}'_{\phi_e,\tau_{\Delta s}}$, as well as the temporal slot embeddings $\mathbf{t}_{\phi_e,\tau_{\Delta t}}, \mathbf{t}'_{\phi_e,\tau_{\Delta t}}$. The updating functions for the l-th layer of our base model (HoST-GNN$_{\text{base}}$) are given by:

$$\mathbf{h}_{vv'} = \mathbf{W}_1^{(l)}\mathbf{h}_v^{(l-1)} \parallel \mathbf{W}_1^{(l)}\mathbf{h}_{v'}^{(l-1)} \parallel \mathbf{y}_{\phi_e} \parallel \mathbf{s}_{\phi_e,\tau_{\Delta s}} \parallel \mathbf{t}_{\phi_e,\tau_{\Delta t}} \tag{9}$$

$$\alpha_{vv'} = \underset{v' \in \mathcal{N}_v}{\text{softmax}}(\sigma(\mathbf{a}^{(l)} \cdot \mathbf{h}_{vv'})) \tag{10}$$

$$\mathbf{h}_v = \sum_{v' \in N_v} \alpha_{vv'}(\mathbf{W}_1^{(l)}\mathbf{h}_{v'}^{(l-1)} + \mathbf{y}'_{\phi_e} + \mathbf{s}'_{\phi_e,\tau_{\Delta s}} + \mathbf{t}'_{\phi_e,\tau_{\Delta t}}) \tag{11}$$

$$\mathbf{h}_v^{(l)} = \tanh(\mathbf{W}_2^{(l)}[\mathbf{h}_v^{(l-1)} || \mathbf{h}_v]) \tag{12}$$

where $\mathcal{N}_v$ is the neighbor set of v, $\mathbf{h}_v^{(l-1)}, \mathbf{h}_{v'}^{(l-1)} \in \mathbb{R}^d$ are node embeddings of v, v' from the $l-1$-th layer, $\mathbf{W}_1^{(l)} \in \mathbb{R}^{d\times d}, \mathbf{W}_2^{(l)} \in \mathbb{R}^{d\times 2d}$, $\mathbf{a}^{(l)} \in \mathbb{R}^{1\times 5d}$ are parameters in the l-th layer, σ is an activation function and $\parallel$ is concatenation. Equation (9) concatenates the embeddings of v, v', the edge type embedding and slot embeddings to better guide the attention calculation based on the heterogeneous spatial-temporal context. Equation (10) is the standard attention coefficient calculation. Equation (11) incorporates the heterogeneous spatial-temporal context into message propagated from v'. Equation (12) combines the aggregated messages $\mathbf{h}_v$ and the previous layer embedding $\mathbf{h}_v^{(l-1)}$ to find potential cross-layer correlations and obtain the updated node embedding for the current layer. The proposed HoST-GNN$_{\text{base}}$ architecture can be easily transformed into the multi-head setting according to Equation (3).

HoST-GNN Layer. As illustrated in Fig. 1, human behavior patterns naturally exist in multiple temporal granularities. To further capture the multi-granular temporal patterns, we introduce multi-granular temporal slot embeddings based on $\mathcal{T}_p$ and $\mathcal{T}'_p$, where $p \in \mathcal{P}_{\mathcal{E}}$. We define a set of N_g temporal granularities $\Omega_{\mathcal{T}} = \{\partial t_\omega\}_{\omega=1}^{N_g}$, where ∂t_ω denotes temporal interval. Then we set a temporal perception range ΔT to determine the number of slots $N_{\mathcal{T}}^{\omega} = \lceil \Delta T / \partial t_\omega \rceil + 1$ and the slot mapping function $\tau_{\mathcal{T}}^{\omega}$ for each granularity ω. For each granularity level ω, we have $\mathcal{T}_{p,\omega} = \{\mathbf{t}_{p,\omega,n}\}_{n=1}^{N_{\mathcal{T}}^{\omega}}, \mathcal{T}'_{p,\omega} = \{\mathbf{t}'_{p,\omega,n}\}_{n=1}^{N_{\mathcal{T}}^{\omega}}$. The full model HoST-GNN integrates the multi-granularity temporal information into HoST-GNN$_{\text{base}}$ by replacing $\mathbf{t}_{\phi_e,\tau_{\Delta t}}, \mathbf{t}'_{\phi_e,\tau_{\Delta t}}$ in Equation (9) and (11) with:

$$\mathbf{t}_{\phi_e,*,\tau_{\Delta t}^{\omega}} = \|_{\omega=1}^{N_g} \mathbf{t}_{\phi_e,\omega,\tau_{\Delta t}^{\omega}} \qquad \mathbf{t}'_{\phi_e,*,\tau_{\Delta t}^{\omega}} = \sum_{\omega=1}^{N_g} \mathbf{t}'_{\phi_e,\omega,\tau_{\Delta t}^{\omega}} \tag{13}$$

and adjusting the dimension of $\mathbf{a}^{(l)}$ in Equation (10) to $\mathbb{R}^{1\times(4+N_g)d}$.

Output Module. To predict the next POI for the target user u and time T, we concatenate its embeddings from all layers of HoST-GNN extracted from $\mathcal{G}_{ego}(u, T)$ as the final embedding:

$$\mathbf{h}_u = \|_{l=0}^{L} \mathbf{h}_u^{(l)} \tag{14}$$

Table 1: Dataset Statistics

Dataset	#User	#POI	#Check-in	Median Δt	Median Δd
Gowalla	17490	120967	2504640	7.34h	2.87km
Foursquare	17025	91614	1621437	24.29h	2.05km

where L is number of layers and $\mathbf{h}_u^{(0)}$ is the initial embedding vector. We employ Multi-Layer Perceptron (MLP) with softmax as the final function for prediction, and solely use $\mathbf{h}_u$ as its input:

$$\hat{\mathbf{y}} = \text{MLP}(\mathbf{h}_u) \tag{15}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{L}|}$ is the predicted probability vector for POIs.

It's noteworthy that our HoST-GNN is flexible and can be easily extended with additional message decorator to be aware of more auxiliary information.

3.3 Objective Function

For each $\hat{\mathbf{y}}$ form Equation (15), we have a cross-entropy based loss:

$$f = -[y_{l'} \log(\hat{y}_{l'}) + \sum_{l \in \mathcal{L}\setminus\{l'\}} (1 - y_l)(1 - \log(\hat{y}_l)] + \lambda\|\Theta\|_2 \tag{16}$$

where l' is the ground-truth location; $\hat{y}_l = \hat{\mathbf{y}}[l]$ is the predicted probability for location l; y_l is the smoothed ground-truth label [36]. Θ denotes all parameters; $\|\cdot\|_2$ is L_2 norm; λ is the coefficient.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets. We evaluate HoST on two public benchmark datasets: Gowalla [3] and Foursquare [45]. We extract worldwide check-in records from Nov. 2009 to Oct. 2010 in Gowalla and from Apr. 2013 to Mar. 2014 in Foursquare. Considering that some sequential methods have relatively strict requirements for sequence length (e.g., LSTPM [35]), for a fair comparison, we only keep users with no less than 50 check-in records, and remove POIs visited less than 10 times in the datasets. We sort the remaining check-in sequences in ascending chronological order, and split the data into train/valid/test sets by 70%/10%/20%. The POIs not occurred in the training set are ignored for evaluation. The detailed statistics are shown in Table 1, where the last two rows show the median value of temporal and spatial intervals between consecutive check-in records.

Baseline Methods. We compare the proposed HoST with three groups of baselines: **(1) Traditional Methods.** Three traditional baselines are adopted, i.e., TOP, U-TOP and MF [18]. TOP and U-TOP are frequency-based ranking methods. TOP adopts global frequencies of POIs in the full training set and recommend the most popular POIs, while U-TOP uses local frequencies based on each user u's historical check-in sequence C_u. MF is a classical collaborative filtering method widely used in recommendation systems. **(2) Sequential Methods.** We adopt basic LSTM [11], spatial-temporal-aware RNN variants ST-LSTM [17], STGCN [50], LSTPM [35], Flashback [44] and attention-based method STAN [26] for this group. ST-LSTM and STGCN are both LSTM variants. ST-LSTM integrates spatial and temporal intervals between consecutive check-in records into LSTM gates, while STGCN introduces additional spatial/temporal gates and cell state. LSTPM is a hierarchical LSTM encoder with a

Table 2: Overall performance. Boldfaced and underlined scores are the best and the second-best ones respectively.

	Gowalla				Foursquare			
	Acc@1	Acc@5	Acc@10	MRR	Acc@1	Acc@5	Acc@10	MRR
TOP	0.0018	0.0073	0.0138	0.0062	0.0033	0.0116	0.0175	0.0086
U-TOP	0.1469	0.3451	0.4224	0.2378	0.2185	<u>0.5437</u>	<u>0.6400</u>	0.3586
MF	0.0162	0.0896	0.1513	0.0249	0.0345	0.2066	0.3259	0.0900
LSTM	0.0899	0.2077	0.2629	0.1480	0.1313	0.3369	0.4056	0.2150
ST-LSTM	0.0892	0.2043	0.2580	0.1460	0.1198	0.3078	0.3826	0.2072
STGN	0.0931	0.2138	0.2701	0.1526	0.1216	0.3191	0.3967	0.2129
LSTPM	0.1353	0.2740	0.3144	0.2103	0.1878	0.4327	0.4998	0.2961
Flashback	0.1782	0.3893	<u>0.4761</u>	0.2776	0.1995	0.5189	0.6250	0.3402
STAN	0.1295	0.2755	0.3198	0.2010	0.1627	0.3494	0.4819	0.2655
LightGCN	0.0247	0.0754	0.1108	0.0326	0.0445	0.1349	0.1968	0.0745
STP-UDGAT	0.1358	0.3130	0.3918	0.2211	0.1615	0.4384	0.5465	0.2868
GETNext	0.2025	0.3617	0.3924	0.2735	0.1866	0.3583	0.3861	0.2626
HMT-GRN	<u>0.2291</u>	<u>0.3899</u>	0.4458	<u>0.3072</u>	<u>0.2780</u>	0.4565	0.5125	<u>0.3639</u>
HoST	**0.3382**	**0.4388**	**0.4843**	**0.3885**	**0.5006**	**0.6181**	**0.6589**	**0.5560**
Improvement	47.62%	11.68%	1.72%	26.47%	80.07%	13.68%	2.95%	52.79%

nonlocal network and a geo-dilated network. Flashback is an RNN-based self-attention method that leverages spatial temporal aware attention to aggregate hidden states of the basic RNN model. We use the reported best RNN-based variant for Flashback here. STAN is an attention-based model that leverages relative spatial-temporal information among non-adjacent visits. **(3) Graph-Augmented Methods.** LightGCN [10], STP-UDGAT [24], GETNext [46] and HMT-GRN [23] are adopted for this group. LightGCN is a popular graph-based collaborative filtering method. STP-UDGAT is a graph-based method that extends GAT to leverage multi-faceted information within multiple customized global POI graphs. GETNext is a Transformer-based method that utilizes general movement patterns from a user-agnostic POI transition graph to refine input embedding and output probability map. Due to the absence of category information, the category related modules of GETNext are removed. HMT-GRN is a graph-enhanced LSTM model that leverages customized spatial and temporal POI graphs to augment LSTM gates, which is trained under a hierarchical region aware multi-task learning framework.

Evaluation Metrics. Following [46], we use the standard Accuracy@K (Acc@K) and Mean Reciprocal Rank (MRR) as metrics. Acc@K reflects the overall performance of top-K recommendation, and MRR shows the quality of ranking.

Implementation Details. We implement our HoST base on PyTorch-Geometric [6]. For a fair comparison, we fix the embedding size $d = 64$ for all methods. We set the hop number $K = 2$ and the edge number $M = 50$ when sampling ego-graphs. For both datasets, we set the spatial granularity ∂s = 0.1km, spatial slot number $N_{\mathcal{S}} = 100$, and the temporal granularity set as $\Omega_{\mathcal{T}}$ ={3h, 12h}. The overall temporal perception range ΔT for Gowalla and Foursquare are 8 weeks and 24 weeks respectively. We stack 2 HoST-GNN layers with hidden dimension 32, and let $\mathcal{E}^{f}_{trans}$ and $\mathcal{E}^{b}_{trans}$ share the temporal and spatial slot embedding. The output module is a 2-layer MLP with LeakyReLU activation function and 128 hidden dimension. As for training, we set $\lambda = 1e-4$, and use dropout with a rate of 0.3 for each layer. We use Adam optimizer [15] with the initial learning rate 0.001 and an exponential decay rate of 0.4 for every 10 epochs. The model is trained for 40 epochs with 128 batch size. We will release the code upon publication.

As for baseline methods, we use the same settings where applicable for MF, LSTM and LightGCN. For ST-LSTM and STGN , we follow the setting in [24], such that in each step, we use the spatial/temporal interval from the previous step to the current step as input. For Flashback, we change the center time from the timestamp of the last check-in of historical visits into the future timestamp to be consistent with our setting. For STAN, considering that training STAN on the whole dataset is extremely memory-consuming due to the large number of users and POIs, we randomly sample 1% of historical check-in sequences to train the model and test model performance, and report the average performance for 20 runs. For the other recent works, we follow the recommended settings in their paper accordingly, except for the embedding size.

4.2 Main Results

We report the evaluation results of our proposed HoST model and the baselines in Table 2. The relative improvement compared with the best baseline is also computed. Following observations and analysis can be obtained from the results: (1) The results on both datasets show that our HoST consistently outperforms other state-of-the-art baseline methods on all metrics, especially on Acc@1 and MRR, which demonstrate the effectiveness of our proposed method. (2) Frequency-based method U-TOP is a very competitive baseline, which reflects that users tend to visit their frequently visited places. (3) MF and LSTM only rely on sparse user-POI interaction matrix to learn user preference, whose performance is not satisfying. ST-LSTM, STGN, LSTPM, STAN, and Flashback incorporate spatial-temporal factors into user preference learning, and achieve performance improvement. However, compared with Flashback, the other sequential baselines are consistently inferior, for that they fail to leverage more informative relative spatial-temporal information. (4) Compared with vanilla sequential methods, STP-UDGAT, GETNext and HMT-GRN attempt to leverage higher-order

Table 3: Ablation study on Foursquare.

Methods	Acc@1	Acc@5	Acc@10	MRR
HoST	**0.5006**	**0.6181**	**0.6589**	**0.5560**
w/o interaction graph	0.4831	0.6133	0.6561	0.5436
w/o transition graph	0.2537	0.5320	0.6149	0.3766
w/o edge direction	0.4062	0.5813	0.6340	0.4866
w/o multi-granularity	0.4961	0.6155	0.6567	0.5520
w/o spatial slot	0.4079	0.5711	0.6252	0.4829
w/o temporal slot	0.2633	0.5249	0.6072	0.3795
w/o edge type	0.4124	0.5848	0.6373	0.4917
w/ Transformer attention	0.4829	0.6093	0.6552	0.5419
w/ same bias embeddings	0.4986	0.6164	0.6585	0.5540
w/ multi-spatial	0.4967	0.6169	0.6583	0.5530

user-POI collaborative information by constructing global graphs, and achieves performance improvement compared with vanilla sequential methods. HMT-GRN and GETNext further develop multi-task learning frameworks to provide more supervision signals, and achieve superior performance compared with STP-UDGAT. However, the state-of-the-art graph-based method HMT-GRN is not consistently superior to Flashback on all metrics. We believe the reason is that Flashback can leverage fine-grained relative spatial-temporal information and dependencies among non-successive check-in records.

4.3 Analysis

Ablation Study. To empirically demonstrate the effectiveness of each proposed component and evaluate its impact on the final performance, we conduct a series of ablation experiments on the Foursquare dataset. Specifically, we design 3 branches of experiments: **(1) HoST-Graph ablation studies**: We first remove the interaction graph (w/o interaction graph) and transition graph (w/o transition graph) in turn. Then attempt to study the effectiveness of edge directions by removing the edge directions (w/o edge direction). **(2) HoST-GNN ablation studies**: We remove the multi-granular temporal embedding (w/o multi-granularity), spatial slot embedding (w/o spatial slot), temporal slot embedding (w/o temporal slot) and edge type embedding (w/o edge type) respectively. (3) **Other ablation studies**: First, we replace the GAT-style attention in HoST-GNN with Transformer attention (w/ Transformer attention). Then, we share the parameters of attention bias embedding and representation embedding for edge type and spatial/temporal slot embeddings (w/ same bias embeddings). At last, we apply the multi-granularity setting to spatial information, and define the spatial perception range and granularity set as ΔS=10km and Ω_S={0.1km,0.5km} respectively (w/ multi-spatial).

Based on the results in Table 3, we can find that the full model achieves the best performance on all metrics, and we have the following observations: *For HoST-Graph*: (1) The transition graph significantly contributes to the final performance improvement, while the interaction graph has much less impact. The reason is that users' co-occurrence does not necessarily mean similar preferences. (2) Distinguishing different information flow among the user and POI nodes is helpful to capture fine-grained semantics. *For HoST-GNN*: (1) The multi-granular temporal slot embedding is beneficial, which demonstrates the necessity of multi-granular temporal information encoding. However, multi-granular spatial information can't provide more informative knowledge. (2) The proposed spatial and temporal slot embedding can both significantly improve model performance, while temporal information is much more important, for that temporal information can reflect sequential information and inherent periodic patterns. (3) The introduction of edge type embedding can help HoST-GNN to capture complex heterogeneous semantics, and further improve the model performance. *For other ablation studies*: (1) The GAT attention mechanism is more expressive than Transformer attention in our model. (2) Sharing the parameters of attention bias embedding and representation bias embedding will cause slight performance degradation. (3) Multi-granular spatial slot embeddings do not have the expected improvements, indicating that the spatial dimension does not have clear multi-granular patterns.

Table 4: Performance for the normal and inactive user group on the Foursquare Dataset.

Method	User Group	Acc@1	Acc@5	Acc@10	MRR
HMT-GRN	Normal	0.2369	0.3448	0.3936	0.2925
	Inactive	0.1338	0.2115	0.2589	0.1766
HoST	Normal	0.2997	0.4283	0.4731	0.3616
	Inactive	0.1892	0.2876	0.3355	0.2395

Cold-Start Performance. We have demonstrated the superiority of HoST in the learning preference of active users with no less than 50 check-ins. However, the majority of inactive users are affected by data sparsity issues most. To further verify the effectiveness of our model for handling inactive users, we conduct extensive experiments on HoST and the most competitive baseline HMT-GRN. We reprocess the Foursquare dataset to keep users with less than 50 check-in records, and further divide them into two groups: *inactive* user group for users with less than 10 check-in records, and *normal* user group for the rest. The POI setting remains unchanged. After splitting these data into new train/valid/test sets, we integrate them into the original ones and retrain the model. The evaluation results for both groups of users are presented in Table 4. Compared with HMT-GRN, our method can significantly boost the performance of both inactive and normal user group, which demonstrate the effectiveness of our HoST to alleviate data sparsity issue. However, compared with the performance gain in active users, the relative improvement on both inactive user groups is much lower for HoST, which reflects that our HoST is better at processing long sequences.

Time and Memory Efficiency. To quantitatively understand the inference efficiency of our method, we select several representative baselines, and test the time and GPU memory consumption of our method and the selected models. We use the average model inference time in seconds for each step as the evaluation metric for time efficiency. All the experiments are conducted with the same batch size and the same device. The results are recorded in Figure 3. Note that we replace the original GCN implementation in GETNext with the sparse version in order to run the test, and the time and memory consumption of STAN is tested based on the whole dataset. Besides, the time consumption of HoST has included the average ego-graph sampling time consumption for

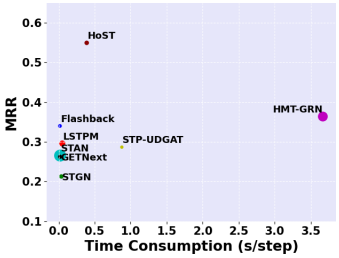
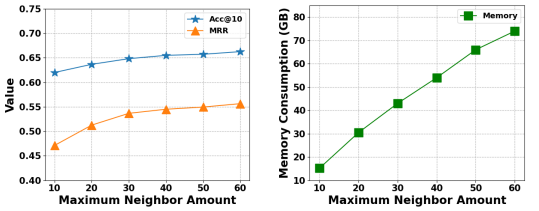


Figure 3: Time and Memory Efficiency Comparison on Foursquare. The area of the circles represents the relative GPU memory consumption of the corresponding method.



(a) Performance Statistics (b) Memory Consumption Statistics

Figure 4: Impact of maximum neighbor amount M

each check-in. For time efficiency, compared with graph-based methods like STP-UDGAT and HMT-GRN, the efficiency of HoST is of great advantage. As for memory efficiency, the GPU memory consumption of HoST is close to many sequential methods with low memory requirements like Flashback, and much lower than HMT-GRN. There are two reasons for this phenomenon: (1) our proposed STC paradigm directly inference on the sampled ego-graphs, which is a one-pass process, while HMT-GRN adopts vanilla SAG paradigm and performs augmentation in each step, which is more time-consuming; (2) HoST only need to load relevant sub-graph structure due to the ego-graph sampling strategy, which will save much memory, while other graph-based methods need to conduct inference on the whole graph, which will repeat even for each time step if the basic sequential model is RNN-based. Overall, our model has competitive inference efficiency compared with existing graph-based methods.

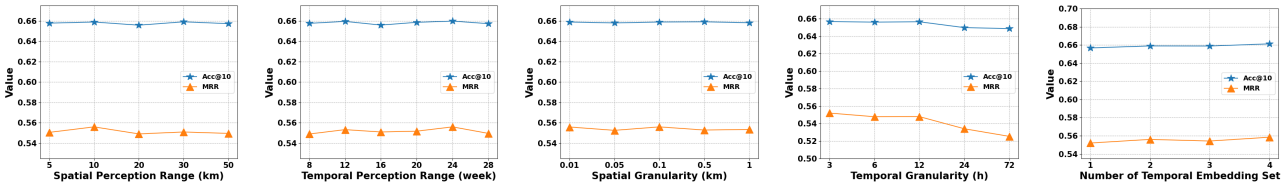
Hyperparameter Sensitivity. To further understand how different data and model architecture setting affect final model performance, we conduct hyperparameter sensitivity analysis for several critical hyperparameters related to ego-graph sampling and spatial-temporal information modeling. We first explore the effect of maximum neighbor number M in HoST-Graph. The evaluation results and corresponding memory consumption are presented in Figure 4(a) and Figure 4(b) respectively. In general, model performance improves with the neighbor amount increasing. However, the performance gain from increasing neighbor amount degrades after M reaches 30, while the memory consumption still grows steadily due to the requirement of loading all pre-computed ego-graphs into memory in advance. Thus, we choose $M = 50$ in previous experiments as a trade-off.

To understand the impact of different spatial-temporal slot settings, we test the sensitivity of HoST-GNN to spatial/temporal perception range and corresponding granularity selection. We first analyze the impact of spatial and temporal perception range ΔT and ΔS, and the evaluation results are presented in Figure 5(a) and Figure 5(b) respectively. The variation of spatial and temporal perception range has no significant influence on predictive accuracy. However, from the trend of MRR curve in both figures, we can find that a shorter spatial range is more beneficial, which may be the result of the regional cluster effect; while a longer temporal range is preferred, which is more helpful to capture long-range periodic patterns. Then, we analyze the impact of granularity selection. On one hand, The evaluation results for spatial granularity ∂s are presented in Figure 5(c), and we can find that different spatial granularity choice has no obvious impact on final model performance. On the other hand, to comprehensively understand the effect of different temporal granularity combinations, we first remove the multi-granular setting and test the importance of each temporal granularity, and then progressively stack the best performing granularities to observe the influence of the size of $\Omega_{\mathcal{T}}$. The evaluation results under single granularity setting are presented in Figure 5(d). Compared with spatial dimension, the choice of temporal granularity can significantly impact predictive accuracy. Fine-grained temporal information is more informative, and the model performance steadily degrades along with the coarser granularity choice. According to the performance ranking, we stack {3h, 12h, 6h, 24h} into $\Omega_{\mathcal{T}}$ in turn and obtain the evaluation results in Figure 5(e). From the experimental results, we find that the model performance is steadily improved along with the increasing number of stacked temporal slot embeddings, and diverse temporal granularity choice will be more helpful for multi-granular temporal information encoding.

5 RELATED WORK

5.1 Next POI Recommendation

The objective of next POI recommendation is to recommend a ranked list of POIs that users most likely visit next based on historical check-in records. Mainstream solutions tend to focus on the sequential influence of individual historical check-in sequences and adopt sequential models. Markov Chains are utilized in earlier studies [2, 5]. Recently, Researchers propose variants of RNNs to capture long- and short-term sequential patterns and further find their correlations with spatial-temporal information. ST-RNN [25], ST-LSTM [17] and STGN [50] attempt to integrate spatial/temporal intervals between consecutive check-ins into RNN and LSTM architecture. LSTPM [35] further uses hierarchical LSTM encoders to capture sub-sequence level knowledge with a context-aware non-local network. Besides, an attention mechanism is also adopted recently to capture the dependencies among non-consecutive check-ins [4, 26, 44]. For instance, Flashback [44] uses a spatial-temporal interval aware attention module to aggregate the output hidden states of RNN. Some recent works also adopt Transformer for next POI recommendation [29, 34, 46].



(a) Spatial Perception Range (b) Temporal Perception Range (c) Spatial Granularity (d) Temporal Granularity (e) Multi-Granular Effect

Figure 5: Impact of spatial-temporal perception range and granularity selection.

Overall, existing sequential methods only learn from highly sparse user-POI interaction data. To alleviate the data sparsity issue, recent graph-augmented works propose to leverage global user-POI collaborative information to enhance sequential methods. We review these works from two aspects: graph construction and model design. For graph construction, [20, 23, 24, 46] defines proximity rules based on spatial, temporal, and transition factors and constructs multiple POI graphs; [8] and [30] leverages user-POI knowledge graph constructed from user-POI interactions and other auxiliary data. As for model design, STP-UDGAT [24] extends GAT to extract multi-faceted information from different graphs and linearly fuse them for prediction. SGRec [20] and HMT-GRN [23] extend RNN architecture to learn from graph-augmented sequences. Graph Flashback [30] and GETNext [46] refine the input or output of the sequential model with features extracted from graphs. [20, 23, 46] further design different multi-task learning frameworks to provide more supervision signals.

Although existing SAG methods can effectively alleviate data sparsity issues, we argue that they fail to model the fine-grained global spatial-temporal context and the multi-granular spatial-temporal dynamics.

5.2 Heterogeneous Graph Neural Networks

Real-world graphs are heterogeneous in nature (e.g., various node/edge types, attributes). With the success of Graph Neural Networks (GNNs) on homogeneous networks [16, 40], they have been extended to heterogeneous graphs to capture complex information. R-GCN [32] treats each edge type as a view and conduct intra-view and inter-view neighborhood aggregation, and CompGCN [38] adopts relation embedding for each edge type to enrich node embeddings. HAN [41] and MAGNN [7] adopt pre-defined meta-paths to define different views. GTN [48] and HetGNN [49] could automatically discover informative meta-paths to avoid using expert knowledge. HTG [12] extends Transformer [39] to heterogeneous graphs. Recently, [27] points out that with proper modification, classic GAT [40] could outperform the SOTA complex heterogeneous graph neural networks. Therefore, rather than design a complex model to learn from heterogeneous information, we build a simple model based on the vanilla GAT.

5.3 Spatial-Temporal Graph Neural Networks

Spatial-temporal graphs appear in various real-world applications, such as traffic monitoring [47], environmental monitoring [14], and link prediction [42]. Generally, there are two kinds of spatial-temporal graphs: discrete and continuous graphs. The discrete graphs are snapshots taken at certain time steps, while an interaction in continuous graphs could appear at any time, For discrete graphs, the dominating practice is to combine sequential models with discrete GNN encoders. DCRNN [21], STGCN [47], EvolveGCN [28] and NET^3 [14] use RNN/CNN as sequential models and graph convolutional networks to encode graphs. DysSAT [31] leverages temporal and structural self-attentions to model sequential dynamics and graph structures. Recently, some methods devote to continuous spatial-temporal graph learning. DyRep [37] and JODIE [19] regard continuous graphs as link streams, and use the temporal point process and RNN to model the graph evolution process respectively. CAW [42] leverages causal anonymous temporal random walk to perform inductive learning of graph evolution patterns. Our constructed graphs are continuous, where the timestamp associated with each edge can be any valid time. The time is mapped to temporal slots in HoST-GNN and each slot has an embedding.

6 CONCLUSION

In this paper, we introduce a novel heterogeneous spatial-temporal method (HoST) to address three under-explored challenges for the next POI recommendation: *(C1) How to construct fine-grained spatial-temporal context graphs? (C2) How to build a unified spatial-temporal model? (C3) How to model multi-granular temporal periodicity?* HoST is comprised of a graph construction method HoST-Graph for *(C1)* and a graph neural network HoST-GNN for *(C2)(C3)*. HoST-Graph first constructs a global heterogeneous spatial-temporal graph by exploiting users' check-in sequences, and then samples an ego-graph for the target user and time to capture the relative spatial-temporal context. HoST-GNN is a unified model which extracts user embeddings and makes recommendations directly based on ego-graphs without intentionally using users' long check-in history. HoST-GNN models the heterogeneous spatial-temporal context by introducing edge type embeddings and spatial-temporal slot embeddings into GAT, and it further captures the multi-granular temporal dynamics via the multi-granular temporal slot embeddings. Comprehensive evaluations on two real-world benchmark datasets demonstrate that our HoST framework is superior than existing methods and it is able to alleviate data sparsity issues for inactive users.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[2] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where You like to Go next: Successive Point-of-Interest Recommendation. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13)*. 2605–2611.

[3] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-Based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. 1082–1090.

[4] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference (WWW '18)*. 1459–1468.

[5] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for next New POI Recommendation. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*. 2069–2075.

[6] Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *Proceedings of the 7th International Conference on Learning Representations (RLGM Workshop) (ICLR '18)*.

[7] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. MAGNN: Metapath Aggregated Graph Neural Network for Heterogeneous Graph Embedding. In *Proceedings of The Web Conference 2020 (WWW '20)*. 2331–2341.

[8] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An Attentional Recurrent Neural Network for Personalized Next Location Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020), 83–90.

[9] Peng Han, Jin Wang, Di Yao, Shuo Shang, and Xiangliang Zhang. 2021. A graph-based approach for trajectory similarity computation in spatial networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 556–564.

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. 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 (SIGIR '20)*. 639–648.

[11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*. 2704–2710.

[13] Shuhui Jiang, Xueming Qian, Jialie Shen, Yun Fu, and Tao Mei. 2015. Author Topic Model-Based Collaborative Filtering for Personalized POI Recommendations. *IEEE Transactions on Multimedia* 17 (2015), 907–918.

[14] Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021. Network of tensor time series. In *Proceedings of the Web Conference 2021*. 2425–2437.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[17] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI'18)*. 2341–2347.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[19] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*. 1269–1278.

[20] Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021. Discovering Collaborative Signals for Next POI Recommendation with Iterative Seq2Graph Augmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 1491–1497. Main Track.

[21] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[22] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. 2009–2019.

[23] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. 2022. Hierarchical Multi-Task Graph Recurrent Network for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1133–1143.

[24] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Jagannadan Varadarajan. 2020. 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 (CIKM '20)*. 845–854.

[25] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. 194–200.

[26] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of the Web Conference 2021 (WWW '21)*. 2177–2185.

[27] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are We Really Making Much Progress? Revisiting, Benchmarking and Refining Heterogeneous Graph Neural Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*. 1150–1160.

[28] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 5363–5370.

[29] Yanjun Qin, Yuchen Fang, Haiyong Luo, Fang Zhao, and Chenxing Wang. 2022. Next Point-of-Interest Recommendation with Auto-Correlation Enhanced Multi-Modal Transformer Network. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 2612–2616.

[30] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. 1463–1471.

[31] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 519–527.

[32] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*. 593–607.

[33] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* (2018).

[34] Lei Shi, Yuankai Luo, Shuai Ma, Hanghang Tong, Zhetao Li, Xiatian Zhang, and Zhiguang Shan. 2023. Mobility Inference on Long-Tailed Sparse Trajectory. *ACM Trans. Intell. Syst. Technol.* 14, 1, Article 18 (jan 2023), 26 pages.

[35] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (2020), 214–221.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[37] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. Dyrep: Learning representations over dynamic graphs. In *Proceedings of the 7th International conference on learning representations (ICLR '19)*.

[38] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR '20)*.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30.

[40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR '18)*.

[41] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference (WWW '19)*. 2022–2032.

[42] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. 2021. Inductive Representation Learning in Temporal Networks via Causal Anonymous Walks. In *Proceedings of the 9th International conference on learning representations (ICLR '21)*.

[43] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.

[44] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudre-Mauroux. 2021. Location Prediction over Sparse User Mobility Traces Using RNNs: Flashback in Hidden States!. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)*. Article 302.

[45] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting User Mobility and Social Relationships in LBSNs: A Hypergraph Embedding Approach. In *The World Wide Web Conference (WWW '19)*. 2147–2157.

[46] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: Trajectory Flow Map Enhanced Transformer for Next POI Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22)*. 1144–1153.

[47] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 3634–3640.

[48] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems* 32 (2019).

[49] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 793–803.

[50] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S. Sheng, and Xiaofang Zhou. 2019. Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (2019), 5877–5884.