Pre-Training Across Different Cities for Next POI Recommendation

KE SUN, School of Computer Science, Wuhan University, China
TIEYUN QIAN*, School of Computer Science, Wuhan University, China
CHENLIANG LI, School of Cyber Science and Engineering, Wuhan University, China
XUAN MA, School of Computer Science, Wuhan University, China
QING LI, Hong Kong Polytechnic University, China
MING ZHONG, School of Computer Science, Wuhan University, China
YUANYUAN ZHU, School of Computer Science, Wuhan University, China
MENGCHI LIU, Guangzhou Key Laboratory of Big Data and Intelligent Education, School of Computer Science, South China Normal University, China

The Point-of-Interest (POI) transition behaviors could hold absolute sparsity and relative sparsity very differently for different cities. Hence, it is intuitive to transfer knowledge across cities to alleviate those data sparsity and imbalance problems for next POI recommendation. Recently, pre-training over a large-scale dataset has achieved great success in many relevant fields, like computer vision and natural language processing. By devising various self-supervised objectives, pre-training models can produce more robust representations for downstream tasks. However, it is not trivial to directly adopt such existing pre-training techniques for next POI recommendation, due to the *lacking of common semantic objects (users or items) across different cities*. Thus in this paper, we tackle such a new research problem of *pre-training across different cities* for next POI recommendation. Specifically, to overcome the key challenge that different cities do not share any common object, we propose a novel pre-training model named CATUS, by transferring the **cat**egory-level **u**niversal tran**s**ition knowledge over different cities. Firstly, we build two self-supervised objectives in CATUS: *next category prediction* and *next POI prediction*, to obtain the universal transition-knowledge across different cities and POIs. Then, we design a *category-transition oriented sampler* on the data level and an *implicit and explicit transfer strategy* on the encoder level to enhance this transfer process. At the fine-tuning stage, we propose a *distance oriented sampler* to better align the POI representations into the local context of each city. Extensive experiments on two large datasets consisting of four cities demonstrate the superiority of our proposed CATUS over the state-of-the-art alternatives. The code and datasets are available at https://github.com/NLPWM-WHU/CATUS.

CCS Concepts: • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Sequential POI recommendation, Pre-training, Sparsity.

*Corresponding author.

Authors' addresses: Ke Sun, sunke1995@whu.edu.cn, School of Computer Science, Wuhan University, Wuhan, Hubei, China; Tieyun Qian, qty@whu.edu.cn, School of Computer Science, Wuhan University, Wuhan, Hubei, China; Chenliang Li, cllee@whu.edu.cn, School of Cyber Science and Engineering, Wuhan University, Wuhan, China; Xuan Ma, yijunma0721@whu.edu.cn, School of Computer Science, Wuhan University, Wuhan, China; Qing Li, qing-prof.li@polyu.edu.hk, Hong Kong Polytechnic University, Hong Kong, China; Ming Zhong, clock@whu.edu.cn, School of Computer Science, Wuhan University, Wuhan, China; Yuanyuan Zhu, yyzhu@whu.edu.cn, School of Computer Science, Wuhan University, Wuhan, China; Mengchi Liu, liumengchi@scnu.edu.cn, Guangzhou Key Laboratory of Big Data and Intelligent Education, School of Computer Science, South China Normal University, Guangzhou, China.

1559-1131/2023/6-ART $15.00
https://doi.org/10.1145/3605554

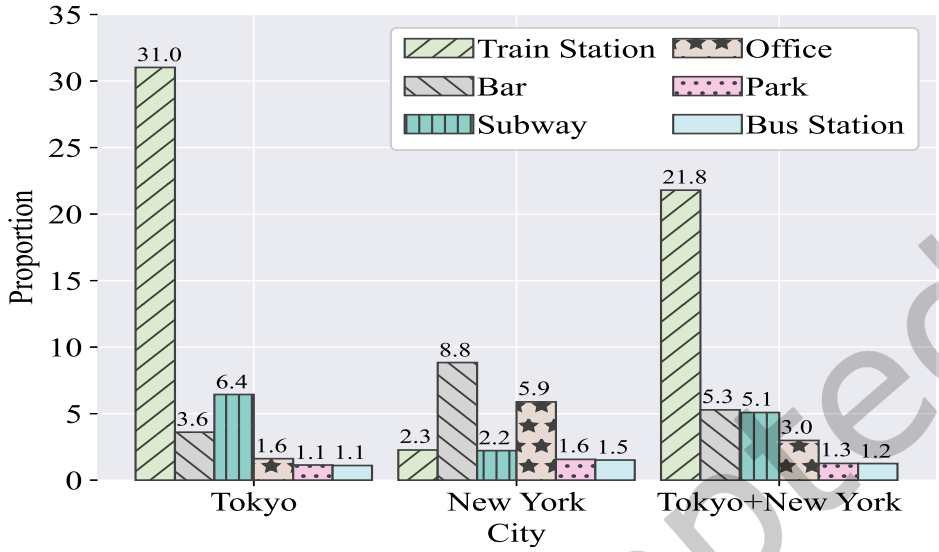


Fig. 1. The proportions of six representative categories of POIs in *Tokyo* and *New York* cities in *Foursquare* dataset. *Tokyo+New York* denotes the proportion of the categories for the combined data of two cities.

1 INTRODUCTION

[2, 30, 32, 36, 38, 40] With the rapid growth of location-based social networks such as Gowalla and Foursquare, identifying next point-of-interest (POI) that will be visited by a user could facilitate many intelligent applications such as personalized location-aware services and public safety monitoring. Therefore, next POI recommendation has drawn substantial attention in the past few years.

Despite the huge volume of historical check-in data generated everyday, data sparsity, as a notorious problem in the recommendation field [33], is still the major obstacle that hinders effective next POI recommendation. Actually, there are two different types of data sparsity problems here: *absolute sparsity* and *relative sparsity*. The absolute sparsity refers to the phenomenon that some kinds of POIs are rarely visited in one city. On the other hand, the relative sparsity indicates that different cities have different distributions for a specific kind of POIs. Figure 1 illustrates the visiting proportions of six representative categories in *Tokyo* and *New York* cities from the Foursquare[1] dataset. It can be seen that POI transitions involving *Park* and *Bus Station* are absolutely sparse in both cities. But it is also obvious that *Train Station* related transitions are relatively much sparser in *New York* than *Tokyo*.

It is intuitive to transfer latent transition-knowledge across cities, which is revealed by the explicit category transitions, for better recommendation since sufficient check-in data can be complementarily accumulated by merging the datasets from different cities. Specifically, the highly skewed distribution can be smoothed to alleviate absolute sparsity problem by learning over different cities together. As shown in Figure 1, the proportion of *Office* related POIs can be increased for *Tokyo* by merging *Tokyo* and *New York* (*i.e.,* from 1.6% to 3.0%). We term this as **knowledge enhancement**. Also, the relative sparsity problem can be addressed by leveraging the transition patterns in other cities. We can learn *Train Station* related patterns from *Tokyo* and then transfer such knowledge

[1]https://sites.google.com/site/yangdingqi/home/foursquare-dataset

to improve the representation learning of the related POIs in *New York* city. We term this **knowledge transfer**. However, simply merging the data of different cities for training and prediction might hurt the modeling of each city, since the merged distribution is different from that of each single city as shown in Figure 1. To this end, we propose to pre-train on the merged data to pursue knowledge transfer across cities, and then fine-tune on each city to fit the city's own unique patterns.

Pre-training technique is a natural yet dominating choice to perform knowledge transfer. The success of pre-training owes much to its self-supervised nature and huge volume of the large-scale corpus which usually can be constructed by merging multiple corpora. The former releases the learning models from labeled data while the latter enables them to learn general knowledge from big data. Inspired by these studies, researchers in the field of recommendation system also propose different pre-training methods for various scenarios [12, 49, 50, 50, 56]. Unlike the corpus in NLP that shares words among multiple corpora, different datasets in recommendation often do not have any common semantic objects (*i.e.,* users or items). Due to this inherent limitation, these existing pre-training methods mainly investigate how to design effective self-supervised objectives. For example, pre-training approaches for next POI recommendation [26, 42] have attempted to exploit the temporal information. Despite the potential feasibility of these temporal pre-training methods on the large dataset, none of them has ever conducted pre-training on the combined dataset from multiple cities. Besides, they are of low utility since the temporal information is a kind of coarse-grained signals. Users may visit different kinds of locations in a very short time period. This limits the learning of intrinsic universal knowledge across cities.

Another line of alternatives is to adopt category information. The reasons for utilizing category information are twofold. Firstly, the POI sets in different cities share the same category set. The category of a POI indicates its functionality, which can be considered as fine-grained semantic information. Secondly, the category transitions reflect the universal patterns across cities. To illustrate this, we plot category transition probabilities among five categories in Figure 2. We observe that users in both cities tend to relax in a *Bar* after eating in a *Noodle House* or work in an *office*, reflecting universal patterns across cities. However, the current technical alternatives mainly perform a conventional parameter-sharing such as utilizing global category embeddings. This simple setting is deficient to pass the transition-knowledge from a universal category to specific POIs in each city. Furthermore, pre-training in cross-domain recommendation [43] depends on the common users in multiple domains, which only constitute a small fraction of all users.

To this end, in this paper we raise a new research problem of pre-training across different cities for next POI recommendation. The main idea is to perform the intrinsic universal knowledge transfer across different cities in a self-supervised manner. To overcome the challenge that different cities do not share any common POIs and users[2], we propose a novel **cat**egory-level **u**niversal tran**s**ition pre-training model, named CATUS. We aim to transfer the universal transition-knowledge via an integration of two self-supervised objectives: *next category prediction* and *next POI prediction*. To further enhance this key purpose, we devise a *category-transition oriented sampler* on the data level and an *implicit and explicit transfer strategy* on the encoder level to pass the knowledge from the universal category to the specific POIs of each city. Specifically, the category-transition oriented sampler performs data augmentation on the data level and generates new semantically similar POI sequences based the same category transitions. Meanwhile, the implicit and explicit transfer strategy works on the encoder level. On one hand, the implicit strategy shares the parameters of POI and category encoders. On the other hand, the explicit strategy guides the POI sequence encoding process with the category relevance. As to the fine-tuning stage, we further extend CATUS to a new version CATUS+DS with a proposed *distance oriented sampler* to better align the POI representations into the local context of each city.

The proposed CATUS for pre-training can work as a plug-in for various recommendation models. We conceptually elaborate the difference between CATUS and the existing counterparts in Figure 3. It is clear that CATUS

[2]There might be some users traveling in multiple cities. However, for the privacy issue, such information is not disclosed in most POI datasets.

(a) Tokyo (b) New York

Fig. 2. Category transition probabilities among five categories in two cities.

can take advantage of multiple datasets from different cities via exploiting the universal knowledge across cities at pre-training stage. Furthermore, CATUS+DS for fine-tuning involves a more effective fine-tuning strategy by integrating local context information at fine-tuning stage. We conduct extensive experiments on two large datasets consisting of four city sub-datasets to verify the effectiveness of CATUS and CATUS+DS. Overall, the main contributions of this paper can be summarized as follows.

- To the best of our knowledge, the proposed CATUS is the first attempt to solve the problem of pre-training across different cities for next POI recommendation. None of existing studies has ever conducted pre-training across different cities.
- We propose a *category-transition oriented sampler* and an *implicit and explicit transfer strategy* under an integration of two self-supervised objectives for univesal transition-knowledge transfer in category level. We further propose to fine-tune POI representations into their own city for better recommendation performance.
- Extensive experiments on two combined large datasets demonstrate the superiority of our proposed model over three state-of-art pre-training models and four typical downstream POI recommendation methods.

2 RELATED WORK

In this section, we first review the literature in next POI recommendation and then investigate the recent progress of pre-training methods in recommendation, especially those in POI recommendation.

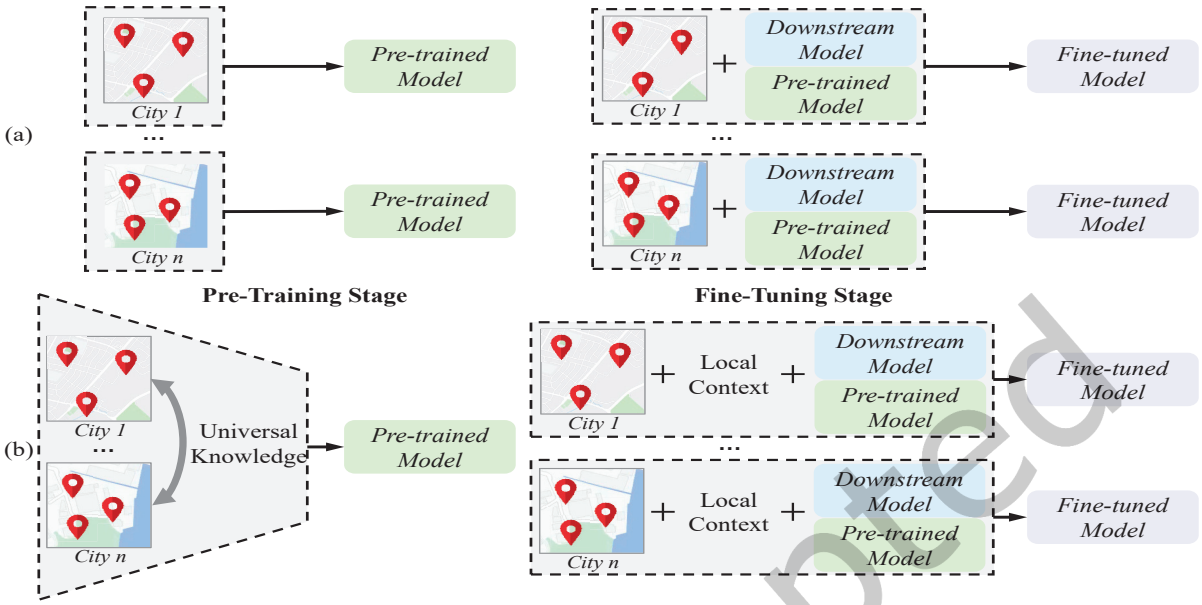


Fig. 3. The comparison between (a) existing methods; (b) our method.

2.1 Next POI Recommendation

Next POI recommendation, as a subtask of sequential recommendation [39], focuses on modeling the transition patterns under spatial-temporal influences and predicts the user's next possible movement. Early researches follow the Markov assumption and explore the short-range behavioral patterns [4, 10, 15, 61]. A representative method is FPMC-LR [4] which factorizes the observed geographical constrained transition matrix. With the prevalence of deep learning, neural networks become pervasive in next POI recommendation, such as recurrent neural networks [11, 19, 27, 31, 37, 60], attention networks [6, 9, 20, 29, 57, 59], and graph neural networks [22, 25, 46, 47, 52]. These neural networks are mainly customized for fitting spatial and temporal contexts. Thus they are able to handle complex long-range transition patterns. For example, Zhao *et al.* [60] develop spatio-temporal gates in LSTM to capture the temporal and spatial impacts on consecutive check-ins. Luo *et al.* [29] extend transformer to model correlations of non-consecutive visits. Lim *et al.* [11] utilize graph attention networks to learn POI-POI relations from both local and global views.

Despite the effectiveness of deep models, POI recommendation still suffers from the data sparsity problem. Limited by the small amount of user-POI interactions, POI recommendation models are unable to fully understand users' visiting preference and easy to get stuck [55]. A promising way to this problem is to leverage auxiliary information like category [14, 55, 58], text content [53], social relations [16]. The auxiliary information can make the data much denser and enable the models to better understand user behaviors.

Recently, meta-learning is also introduced into the field of next POI recommendation for dealing with the sparsity problem [3, 7, 18, 35]. Typical meta-learning methods like MFNP [35], PREMERE [18], and Meta-SKR [7] treat the modeling of a specific user's preferences as an individual task, while CHAML [3] deals with all users' general preference in each city. Another line of research is based on the graph modeling [21, 24, 34]. These methods construct graphs with transition, spatial or temporal relations, so as to enrich POI representations from

the global views. For example, a recent method seq2Graph [21] augments a user's short check-in sequence with other users' sequences to integrate collaborative signals across semantically correlated check-in sequences.

Nevertheless, the POI representations in these methods are randomly initialized and optimized during training, thus they are unaware of valuable latent knowledge across cities.

2.2 Pre-training for Recommendation

To combat the data sparsity problem, researchers resort to the pre-training technique and self-supervised learning in NLP [8, 44] and adopt it into various recommendation tasks. These models are optimized by not only the common recommendation objective but also the self-supervised objectives. These methods can be categorized into two groups: graph modeling [12, 13, 28, 43] and sequence modeling [45, 48–50, 56, 62]. The basic idea is to construct unobserved self-supervised signals for learning useful latent features. For example, Hao *et al.* [12] learn enhanced user embeddings from sampled subgraphs. Yuan *et al.* [56] extract consistency information from artificial noisy sequence data. However, these methods are not suitable for next POI recommendation where a user's movement is severely constrained by her context, such as distance and time.

Several pioneering studies have brought the pre-training technique into next POI recommendation for mitigating the sparsity issue [1, 26, 42]. CTLE [26] applies BERT [8] to the user trajectories and proposes a new masked hour prediction task. TALE [42] incorporates temporal information into the CBOW framework to learn time-aware POI embeddings.

Although the above pre-training models have improved the downstream models' performance, none of them has ever conducted pre-training on the combined large dataset from multiple cities.

3 METHODOLOGY

The proposed CATUS for pre-training consists of two base sequence encoders, two pre-training tasks, a category-transition oriented sampler, and an implicit and explicit strategy. The whole pre-training workflow is illustrated in Figure 4a. The proposed CATUS+DS for fine-tuning consists of a well pre-trained sequence encoder from CATUS, a distance oriented sampler, and a chosen downstream model. The whole fine-tuning workflow is illustrated in Figure 4b. In the following, we describe each component in detail.

3.1 Problem Formulation

Suppose we have a set of cities $\mathcal{Y} = \{y_1, y_2, ..., y_{|\mathcal{Y}|}\}$. For city y_i, let $\mathcal{U}_i$ and $\mathcal{L}_i$ denote the user set and POI set respectively. By merging all the users and POIs from all the cities, we have the union user set $\mathcal{U}$ and POI set $\mathcal{L}$. Each POI $l \in \mathcal{L}$ is associated with a category label $c \in C$ which is shared across cities. Hence, two POIs from different cities might belong to the same category. A check-in record can be seen as a tuple (u, l, t), indicating that user u visited POI l at timestamp t. By sorting a user u's historical check-in records chronologically, we can obtain the POI sequence as $S_l = \{l_1, l_2, ..., l_n\}$, where n is the sequence length. Then, we map each continuous timestamp into 48 discrete time slots to incorporate temporal features following previous works [37], and obtain the corresponding time sequence $S_t = \{t_1, t_2, ..., t_n\}$, where $t_i \in \{0, 1, ..., 47\}$. Similarly, the category sequence can be formed as $S_c = \{c_1, c_2, ..., c_n\}$, where c_i is the category label of l_i in S_l. For each city y_i, we consider the set of all sequences of S_l, S_c, S_t for all users as $\mathcal{D}_i$. The large dataset $\mathcal{D}$ is the union of all city sub-datasets, i.e., $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup ... \cup \mathcal{D}_{|\mathcal{Y}|}$. Given $\mathcal{D}$, the target is to pre-train a unified model, which can generate expressive POI representations and benefit various downstream sequential POI recommendation models for each city y_i. Note that the important notations used in this paper are stated in Table 1. For ease of understanding, we use boldface type to indicate the vector or matrix.

Table 1. Notations in this paper.

Notation	Definitions
$\mathcal{Y}$	the set of cities.
$\mathcal{C}$	the set of categories.
$\mathcal{U}, \mathcal{L}$	the set of users, POIs for all cities.
$\mathcal{U}_i, \mathcal{L}_i$	the set of users, POIs for city y_i.
$\mathcal{D}_i$	the sub-dataset for city y_i.
$\mathcal{D}$	the large dataset for all cities.
$\lvert\mathcal{D}_i\rvert$	the number of POI or category sequence samples from city y_i.
$\lvert\mathcal{D}\rvert$	the number of POI or category sequence samples from all cities.
S_l, S_c, S_t	POI, category, and time sequence.
$\widehat{S}_l, \widetilde{S}_l$	generated POI sequences by the category-transition oriented sampler and the distance oriented sampler.
$\boldsymbol{E}_{i,l}, \boldsymbol{E}_{i,c}$	the embedding matrix for city y_i w.r.t POI and category.
$\boldsymbol{S}_{i,l}, \boldsymbol{S}_{i,c}$	the output matrix of Embedding Layer w.r.t S_l and S_c from city y_i.
$\boldsymbol{S}^M_{i,l}, \widetilde{\boldsymbol{S}}^M_{i,l}$	the output matrix of M-th Self-Attention Layer w.r.t S_l and $\widetilde{S}_l$ from city y_i.
$\boldsymbol{s}_{i,l}, \boldsymbol{s}_{i,c}, \widehat{\boldsymbol{s}}_{i,l}$	the output vector of Encoder w.r.t S_l, S_c, and $\widehat{S}_l$ from city y_i.
$\boldsymbol{s}^j_{i,l}, \widehat{\boldsymbol{s}}^j_{i,l}$	the output vector of Encoder w.r.t S_l and $\widehat{S}_l$ from city y_i, calculated using the category representations from city y_j.
λ	the hyper-parameter for balancing S_l and $\widehat{S}_l$ at the pre-training stage.
α	the hyper-parameter for balancing S_l and $\widetilde{S}_l$ at the fine-tuning stage.
ρ	the proportion of POIs in S_l to be replaced.
PM, DM	the pre-training model and the downstream model.

3.2 Base Sequence Encoder

The sequence encoder is devised as the Transformer [41] module in CATUS, as shown in Figure 5.

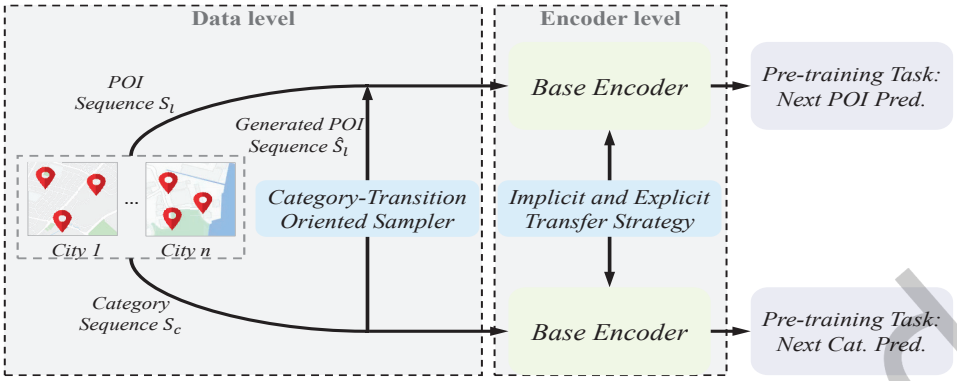
Embedding Layer. For each city y_i, we utilize two embedding matrices $\boldsymbol{E}_{i,l}$ and $\boldsymbol{E}_{i,c}$ for POI and category, respectively. These city specific embeddings ensure the preservation of universal transition-knowledge in that city. For a POI (or category) sequence S_l (or S_c) from y_i, we look up the embedding matrix and obtain an input representation matrix $\boldsymbol{I}_{i,l}$ (or $\boldsymbol{I}_{i,c}$) $\in \mathbb{R}^{n\times d}$, where d is the embedding size. Then we further include the global position embedding matrix $\boldsymbol{P}$ and time embedding matrix $\boldsymbol{T}$ for S_l (or S_c): $\boldsymbol{S}_{i,l} = \boldsymbol{I}_{i,l} + \boldsymbol{P} + \boldsymbol{T}$ (or $\boldsymbol{S}_{i,c} = \boldsymbol{I}_{i,c} + \boldsymbol{P} + \boldsymbol{T}$). Here, $\boldsymbol{P}$ works the same as that in the self-attention mechanisms, which encodes the absolute positions in a sequence. $\boldsymbol{T}$ is mapped via a global time embedding matrix for corresponding time sequence S_t.

Self-Attention Layer. In the transformer based encoder, we stack M self-attention layers. We only present the critical operation of the layer and refer the reader to [41] for more details.

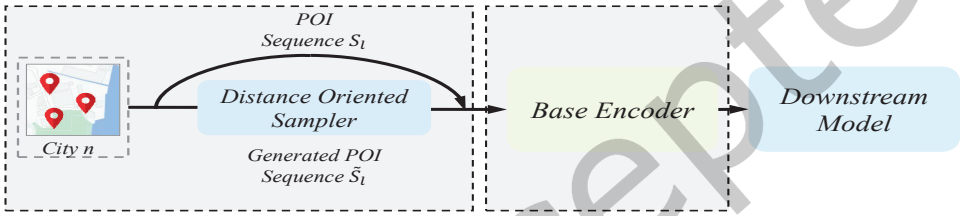
In each self-attention layer, the sequence representation is updated as follows:

$$\text{SelfAtt}(\boldsymbol{S}_{i,*}) = \text{softmax}(\frac{\boldsymbol{S}_{i,*}\boldsymbol{W}^{\boldsymbol{Q}}(\boldsymbol{S}_{i,*}\boldsymbol{W}^{\boldsymbol{K}})^T}{\sqrt{d}})\boldsymbol{S}_{i,*}\boldsymbol{W}^{\boldsymbol{V}} \tag{1}$$

where $\boldsymbol{S}_{i,*}$ is the input for the self-attention layer, $\boldsymbol{W}^*$ denotes the converting matrices for query, key, value. Here $\boldsymbol{S}_{i,*}$ refers to either $\boldsymbol{S}_{i,l}$ or $\boldsymbol{S}_{i,c}$. Apparently, SelfAtt$(\cdot)$ calculates the pairwise affinity within $\boldsymbol{S}_{i,*}$, and outputs



(a) The pre-training workflow of CATUS.



(b) The fine-tuning workflow of CATUS+DS.

Fig. 4. The workflow of our proposed methods.

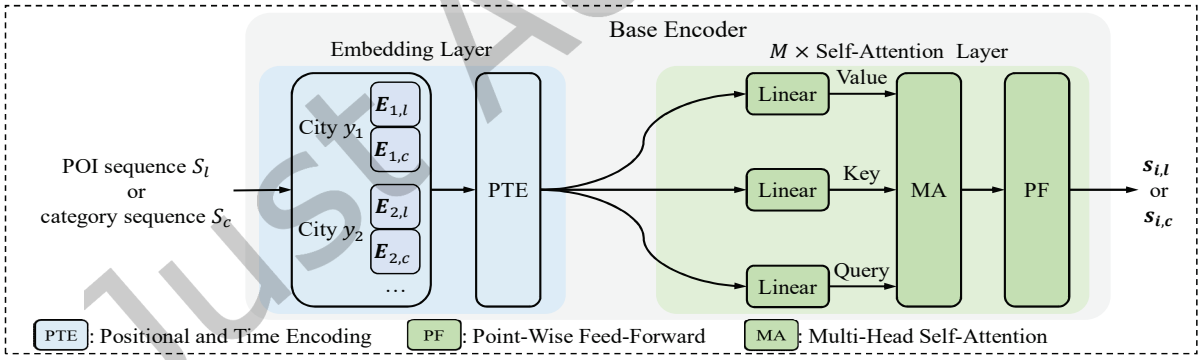
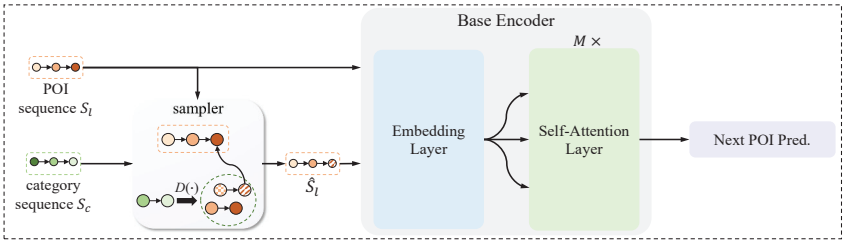


Fig. 5. Details of the base sequence encoder.

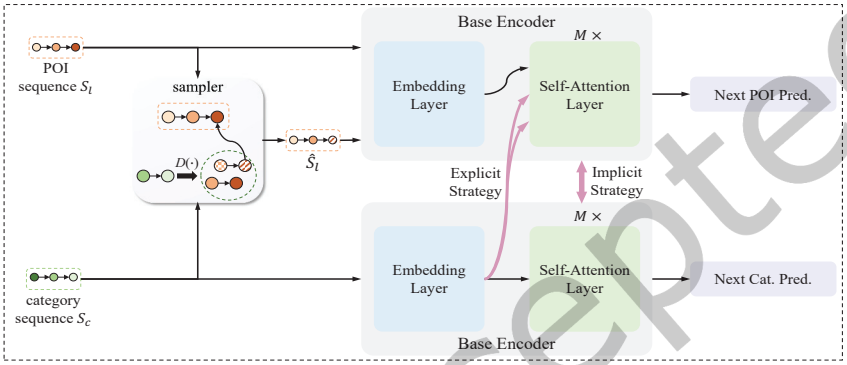
a weighted sum of vectors in $S_{i,*}W^V$. By stacking the self-attention layer for M times, we derive a sequential contextual representation matrix $S^M_{i,*}$, and $S^0_{i,*}$ is equal to $S_{i,*}$. We then take the representation ($s_{i,l}$ or $s_{i,c}$) of the last item in $S^M_{i,*}$ for pre-training.

To summarize, given a POI sequence S_l or a category sequence S_c from y_i, the encoded representation $s_{i,l}$ or $s_{i,c}$ is generated by:

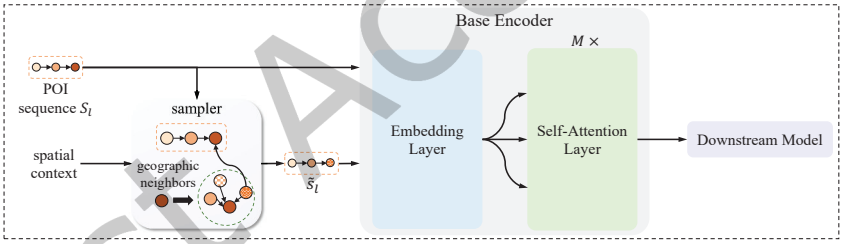
$$s_{i,l} = \text{Encoder}(S_l; \theta_l) \tag{2}$$



(a) Category-transition oriented sampler in CATUS.



(b) Implicit and explicit transfer strategy in CATUS.



(c) Distance oriented sampler in CATUS+DS.

Fig. 6. Details of our proposed framework. (a) The category-transition oriented sampler in CATUS for pre-training. (b) The implicit and explicit transfer strategy in CATUS for pre-training. (c) The distance oriented sampler in CATUS+DS for fine-tuning.

$$\boldsymbol{s}_{i,c} = \text{Encoder}(S_c; \theta_c) \tag{3}$$

where θ_l and θ_c denote the learnable parameter sets of encoders for POI and category sequences, respectively.

3.3 Pre-Training Tasks

To obtain the universal transition-knowledge across different cities and POIs, we adopt two self-supervised objectives: *a next category prediction task* and *a next POI prediction task*.

Next Category Prediction. Given the next POI l_{n+1} to be predicted, we calculate the likelihood score for c_{n+1} by the dot product operation:

$$P(c_{n+1}|S_c) = \boldsymbol{s}_{i,c} \cdot \boldsymbol{e}_{i,c_{n+1}} \tag{4}$$

where $\boldsymbol{e}_{i,c_{n+1}}$ is the embedding for category c_{n+1} of l_{n+1} in city y_i. Then we randomly sample a negative category c^-_{n+1}, $c^-_{n+1} \neq c_{n+1}$, and utilize the pairwise ranking loss as the optimization objective:

$$\mathcal{L}_{CP} = -\log(\sigma(P(c_{n+1}|S_c) - P(c^-_{n+1}|S_c))) \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function.

Next POI Prediction. Similarly, we get the encoded POI representation $\boldsymbol{s}_{i,l}$ of a POI sequence S_l through Eq. 2. Then we predict the next POI l_{n+1}, and construct the objective function as follows:

$$P(l_{n+1}|S_l) = \boldsymbol{s}_{i,l} \cdot \boldsymbol{e}_{i,l_{n+1}} \tag{6}$$

$$\mathcal{L}_{PP} = -\log(\sigma(P(l_{n+1}|S_l) - P(l^-_{n+1}|S_l))) \tag{7}$$

where $\boldsymbol{e}_{i,l_{n+1}}$ is the embedding of l_{n+1} in city y_i, and l^-_{n+1} is a negative POI randomly sampled from the same city y_i.

3.4 Category-Transition Oriented Sampler

The category-transition oriented sampler in the data level performs data augmentation to pass the universal knowledge from categories to POIs, so as to enhance the knowledge transfer process, as shown in Figure 6a. Different from previous augmentation samplers (*e.g.,* POI masking), our sampler keeps new sample $\widehat{S_l}$ semantically consistent with the original one from the viewpoint of category-transition. That is, we replace a POI in S_l with another one from the semantically similar POI transitions belonging to the same category transition, instead of from those belonging to the same category.

Specifically, we first gather POI transitions belonging to the same category transition for each city. Considering a city y_i, given all observed check-in sequences and category sequences from $\mathcal{D}_i$, we build a category-transition dictionary function $D_i(x)$. The function $D_i(x)$ maps a category transition key $c_j \rightarrow c_k$ into a set of corresponding POI transitions $D_i(c_j \rightarrow c_k)$ that occur in city y_i. Each POI transition $l_j \rightarrow l_k$ in $D_i(c_j \rightarrow c_k)$ must meet the following two conditions: (1) l_j and l_k have appeared in at least one POI sequence consecutively; (2) c_j and c_k are the category labels of l_j and l_k respectively.

For a better illustration, here we give a concrete example, in which city y_i has two POI sequences $l_1 \rightarrow l_2 \rightarrow l_3$ and $l_4 \rightarrow l_5 \rightarrow l_6$. We assume they both follow the same category sequence $c_1 \rightarrow c_2 \rightarrow c_3$. Then we can create a category transition dictionary for y_i as follows:

$$D_i(x) = \begin{cases} \{l_1 \rightarrow l_2, l_4 \rightarrow l_5\} & \text{if } x = c_1 \rightarrow c_2 \\ \{l_2 \rightarrow l_3, l_5 \rightarrow l_6\} & \text{if } x = c_2 \rightarrow c_3 \\ \{\} & \text{otherwise} \end{cases} \tag{8}$$

Clearly, all POI transitions in the same set (*e.g.,* $D_i(c_1 \rightarrow c_2)$) follow the same category transition and share the similar interest shift, even though they are totally distinguished from each other.

After building $D_i(\cdot)$ for city y_i, the sampler generates augmented sequences with the replacement operation. Suppose there is an original POI sequence S_l and a parallel category sequence S_c in city y_i, we randomly choose ρ proportion of POIs in S_l to be replaced. For each chosen POI l_m at the m-th position, we obtain $D_i(c_{m-1} \rightarrow c_m)$ by looking up $D_i(\cdot)$ of city y_i with key $c_{m-1} \rightarrow c_m$. Here c_{m-1} and c_m are the category labels of l_{m-1} and l_m respectively. Those POI transitions in $D_i(c_{m-1} \rightarrow c_m)$ are semantically similar to $l_{m-1} \rightarrow l_m$. From this set, the sampler randomly chooses a POI transition $l'_{m-1} \rightarrow l'_m$, and replaces l_m in S_l with l'_m, resulting in a new augmented sequence $\widehat{S_l}$ with $l_{m-1} \rightarrow l'_m$. Following the above example, l_3 in $l_1 \rightarrow l_2 \rightarrow l_3$ could be replaced by l_6.

Although there is a good chance that l_{m-1} and the chosen l'_m have not been visited consecutively, they have played consecutive roles in the same category transition, *e.g.,* l_{m-1} occurs as the preceding POI of c_{m-1} while

l'_m occurs as the succeeding POI of c_{m-1}. Thus these POI sequence samples are semantically consistent and pre-training on them enhances the universal transition-knowledge transfer. One might worry that this replacement may mix the entire context and leads to the loss of geographical features and user preferences. In our work, it does not matter since we only pursue knowledge transfer at the pre-training stage and leave the context modeling to the fine-tuning stage.

At last, the generated sample $\widehat{S}_l$, as well as the original sample S_l, are employed for the next POI prediction task, which leads to the following loss function:

$$\widehat{\mathcal{L}}_{PP} = -\log(\sigma(P(l_{n+1}|\widehat{S}_l) - P(l^-_{n+1}|\widehat{S}_l))) \tag{9}$$

3.5 Implicit and Explicit Transfer Strategy

With the same goal of category-transition oriented sampler, we further develop an encoder level implicit and explicit transfer strategy between POI and category encoders, as illustrated in Figure 6b.

3.5.1 Implicit Transfer Strategy. we share parameters of POI and category encoders, except for embedding matrices. Through this, the POI sequences from each city are encoded under the influence of universal transition-knowledge maintained in the category encoder. Specifically, we have $\theta_l = \theta_c$, excluding $\boldsymbol{E}_{i,l}$ and $\boldsymbol{E}_{i,c}$ in each city.

3.5.2 Explicit Transfer Strategy. The category relevance explicitly reveals the relation between two categories and offers the possibility of a universal transition pattern. Here, we propose to guide the POI sequence encoding with category relevance too.

In detail, for a category sequence S_c, we first look up category embedding matrices (*i.e.,* $\boldsymbol{E}_{1,c}, \boldsymbol{E}_{2,c}, ..., \boldsymbol{E}_{|\mathcal{Y}|,c}$) for all cities. The resultant $|\mathcal{Y}|$ category representations (*i.e.,* $\boldsymbol{S}_{1,c}, \boldsymbol{S}_{2,c}, ..., \boldsymbol{S}_{|\mathcal{Y}|,c}$) are then used for attention calculation in the self-attention layer for POI sequence S_l from city y_i. Taking $\boldsymbol{S}_{j,c}$ from city y_j as an example, we rewrite Equation 1 as follows:

$$\text{SelfAtt}(\boldsymbol{S}_{i,l}, \boldsymbol{S}_{j,c}) = \text{softmax}(\frac{\boldsymbol{S}_{j,c}\boldsymbol{W}^{\boldsymbol{Q}}(\boldsymbol{S}_{j,c}\boldsymbol{W}^{\boldsymbol{K}})^T}{\sqrt{d}})\boldsymbol{S}_{i,l}\boldsymbol{W}^{\boldsymbol{V}} \tag{10}$$

Based on this modification, the encoder outputs a new vector $\boldsymbol{s}^j_{i,l}$ for S_l from city y_i, calculated using the category representation $\boldsymbol{S}_{j,c}$ from city y_j. With the $|\mathcal{Y}|$ category representations from all cities, we obtain a new vector set $\left\{\boldsymbol{s}^1_{i,l}, \boldsymbol{s}^2_{i,l}, ..., \boldsymbol{s}^{|\mathcal{Y}|}_{i,l}\right\}$ for S_l. Then, we utilize them to predict the next POI independently as follows:

$$P^j(l_{n+1}|S_l) = \boldsymbol{s}^j_{i,l} \cdot \boldsymbol{e}_{i,l_{n+1}}, j \in \{1, 2, ..., |\mathcal{Y}|\} \tag{11}$$

$$\mathcal{L}^{y_j}_{PP} = -\log(\sigma(P^j(l_{n+1}|S_l) - P^j(l^-_{n+1}|S_l))) \tag{12}$$

It could be time-consuming to optimize the pairwise objective in Equation 12 for all cities. Besides, not all cities can provide valuable category relevance to S_l. Thus, we select the most informative city and choose the max pairwise loss $\mathcal{L}^y_{PP}$ for optimization:

$$\mathcal{L}^y_{PP} = \max(\mathcal{L}^{y_1}_{PP}, \mathcal{L}^{y_2}_{PP}, ..., \mathcal{L}^{|\mathcal{Y}|}_{PP}) \tag{13}$$

We also apply the explicit strategy to the augmented sequence $\widehat{S}_l$, leading to the set of vectors $\left\{\widehat{\boldsymbol{s}}^1_{i,l}, \widehat{\boldsymbol{s}}^2_{i,l}, ..., \widehat{\boldsymbol{s}}^{|\mathcal{Y}|}_{i,l}\right\}$ and the following objective:

$$\widehat{\mathcal{L}}^y_{PP} = \max(\widehat{\mathcal{L}}^{y_1}_{PP}, \widehat{\mathcal{L}}^{y_2}_{PP}, ..., \widehat{\mathcal{L}}^{|\mathcal{Y}|}_{PP}) \tag{14}$$

Algorithm 1 Framework of CATUS.

Input: The large dataset across cities, $\mathcal{D}$; The hyper-parameter, λ;
Output: optimal encoder parameters;

Randomly initialize encoder parameters;
while not converged **do**
  Sample a batch of category sequences $\mathcal{B}_{S_c}$ from $\mathcal{D}$;
  for $S_c \in \mathcal{B}_{S_c}$: **do**
    Identify the city y_i of S_c and the corresponding embedding matrix $\boldsymbol{E}_{i,c}$;
    Encode S_c to $\boldsymbol{s}_{i,c}$;
    Calculate the next category prediction loss $\mathcal{L}_{CP}$;
  end for
  Update encoder parameters by the gradient descend;
  Sample a batch of POI sequences $\mathcal{B}_{S_l}$ from $\mathcal{D}$;
  for $S_l \in \mathcal{B}_{S_l}$: **do**
    Identify the city y_i of S_l and the corresponding embedding matrix $\boldsymbol{E}_{i,l}$;
    Generate $\widehat{S}_l$ of S_l by the category-transition oriented sampler;
    Encode S_l and $\widehat{S}_l$ to $\boldsymbol{s}_{i,l}$ and $\widehat{\boldsymbol{s}}_{i,l}$, respectively;
    Encode S_l and $\widehat{S}_l$ to $\left\{\boldsymbol{s}_{i,l}^1, \boldsymbol{s}_{i,l}^2, ..., \boldsymbol{s}_{i,l}^{|\mathcal{Y}|}\right\}$ and $\left\{\widehat{\boldsymbol{s}}_{i,l}^1, \widehat{\boldsymbol{s}}_{i,l}^2, ..., \widehat{\boldsymbol{s}}_{i,l}^{|\mathcal{Y}|}\right\}$, respectively;
    Calculate the next POI prediction losses $\mathcal{L}_{PP}$, $\widehat{\mathcal{L}}_{PP}$, $\mathcal{L}_{PP}^y$, and $\widehat{\mathcal{L}}_{PP}^y$;
  end for
  Calculate $\overline{\mathcal{L}_{PP}^y}$ and $\overline{\widehat{\mathcal{L}}_{PP}^y}$;
  Calculate $\mathcal{L}_{PP}^f$;
  Update encoder parameters by the gradient descend;
end while

3.6 Pre-Training

As for model optimization, we develop a two stage-process. At the first stage, we conduct the next category prediction task and directly minimize $\mathcal{L}_{CP}$. At the subsequent stage, we focus on the next POI prediction task and strike a balance among $\mathcal{L}_{PP}$, $\widehat{\mathcal{L}}_{PP}$, $\mathcal{L}_{PP}^y$, and $\widehat{\mathcal{L}}_{PP}^y$. Compared to $\mathcal{L}_{PP}$ and $\widehat{\mathcal{L}}_{PP}$, $\mathcal{L}_{PP}^y$ and $\widehat{\mathcal{L}}_{PP}^y$ are constructed by category relevance from all cities, which may introduce noise. As a result, we give optimization priority to the informative $\mathcal{L}_{PP}$ and $\widehat{\mathcal{L}}_{PP}$ during the pre-training process. Precisely, we first scratch a batch of POI sequences $\mathcal{B}_{S_l}$ from $\mathcal{D}$, and generate a batch of augmented samples $\mathcal{B}_{\widehat{S}_l}$ using the category-transition oriented sampler. Next we calculate the average losses $\overline{\mathcal{L}_{PP}^y}$ and $\overline{\widehat{\mathcal{L}}_{PP}^y}$ over $\mathcal{B}_{S_l}$ and $\mathcal{B}_{\widehat{S}_l}$, respectively. At last, for each sample $S_l \in \mathcal{B}_{S_l}$ and the corresponding augmented sample $\widehat{S}_l \in \mathcal{B}_{\widehat{S}_l}$, we define the final objective for next POI prediction as:

$$\mathcal{L}_{PP}^f = \lambda \max(\mathcal{L}_{PP}, \overline{\mathcal{L}_{PP}^y}) + (1-\lambda)\max(\widehat{\mathcal{L}}_{PP}, \overline{\widehat{\mathcal{L}}_{PP}^y}) \tag{15}$$

where λ is a hyper-parameter to balance the effects of original samples and augmented samples. We summarize the whole pre-training details of CATUS in Algorithm 1.

3.7 Fine-Tuning

At the fine-tuning stage, the pre-trained CATUS is used to generate expressive POI representations which greatly improve the recommendation performance of downstream models for each city. Suppose we have a pre-trained

CATUS model PM and a randomly initialized downstream sequential POI recommendation model DM for city y_i, given a POI sequence S_l, we generate a new representation $\boldsymbol{S}_{i,l}^{M}$ for S_l calculated by the encoder in PM, which is taken as input to DM for fine-tuning with the task of next POI prediction.

Distance Oriented Sampler. At this stage, we further extend CATUS with a distance-oriented sampler for a better fine-tuning, which aligns the pre-trained POI representations into the local spatial context of each city. We name this extended version as CATUS+DS, as shown in Figure 6c. The main idea of the sampler is to augment the observed POI sequences by replacing POIs with their geographical neighbors. This is inspired by the fact that when a user u visits a specific POI l, he/she is likely to visit geographic neighbors of l according to the geographical clustering phenomenon [54]. Specifically, given a POI sequence S_l, we randomly choose ρ[3] proportion of POIs in S_l. Each chosen POI l is replaced with a randomly sampled neighbor from l's top-20 geographic nearest POIs, leading to a new local spatial context-aware sequence $\widetilde{S_l}$. The new sequence $\widetilde{S_l}$ and the original sequence S_l are both treated as training samples for fine-tuning the downstream next POI recommendation models. Suppose the losses of a downstream model are $\mathcal{L}_{DM}$ and $\widetilde{\mathcal{L}}_{DM}$ w.r.t S_l and $\widetilde{S_l}$, we introduce a hyper-parameter α to balance the effects of S_l and $\widetilde{S_l}$ and define the new loss for fine-tuning as:

$$\mathcal{L}_{DM}^{new} = \alpha \mathcal{L}_{DM} + (1-\alpha)\widetilde{\mathcal{L}}_{DM} \tag{16}$$

We present the fine-tuning details of CATUS+DS in Algorithm 2.

Algorithm 2 Framework of CATUS+DS.

Input: The sub-dataset of city y_i, $\mathcal{D}_i$; The hyper-parameter, α; The pre-trained CATUS model, PM; The downstream model, DM

Output: optimal parameters of the downstream model;

```
Randomly initialize parameters of the downstream model;
while not converged do
    Sample a batch of POI sequences B_{S_l} from D_i;
    for S_l ∈ B_{S_l}: do
        Generate S̃_l of S_l by the distance oriented sampler;
        Encode S_l and S̃_l to S^M_{i,l} and S̃^M_{i,l} by PM, respectively;
        Send S^M_{i,l} and S̃^M_{i,l} to DM as pre-trained POI representations;
        Calculate the losses of the downstream model L_DM and L̃_DM;
    end for
    Calculate L^{new}_{DM};
    Update parameters of PM and DM by the gradient descend;
end while
```

3.8 Time Complexity Analysis

In this subsection, we investigate the complexity of proposed CATUS for pre-training and CATUS+DS for fine-tuning. For both of them, the sampler and the self-attention layer induce the main time cost.

Let n denote the length of each POI sequence or category sequence, $|\mathcal{D}|$ denote the number of sequence samples from all cities, and $|\mathcal{D}_i|$ denote the number of sequence samples from city y_i. [4]

[3] ρ here is set to the same value as that in category-transition oriented sampler.

[4] Each POI in a POI sequence belongs to a category in the corresponding category sequence, thus POI and category sequences are the same length. So as the number of sequence samples.

We first analyze the cost of two samplers in CATUS and CATUS+DS, respectively. In CATUS, the category-transition oriented sampler needs to iterate over check-in records from all cities to build the category-transition dictionary function, requiring the time complexity of $O(n|\mathcal{D}|)$. In CATUS+DS, the distance oriented sampler needs to calculate the geographical distance between each pair of POIs in city y_i, leading to a time complexity of $O(|\mathcal{L}_i|^2)$, where $|\mathcal{L}_i|$ denotes the number of POIs in city y_i. Furthermore, when generating new POI samples, *e.g.,* $\widehat{S_l}$ and $\widetilde{S_l}$, these samplers randomly choose ρ proportion of POIs in the original POI sequence S_l to be replaced. Therefore the corresponding time complexity is $O(\rho n|\mathcal{D}|)$ and $O(\rho n|\mathcal{D}_i|)$ w.r.t above two samplers.

Then for the self-attention layer, we assume the numbers of heads and layers are set to one for simplicity, and the complexity of conducting the self-attention operation over one sequence sample is $O(n^2d)$ [41]. In CATUS, this operation is conducted over category samples from all cities for the next category prediction task, hence we have the time complexity of $O(n^2d|\mathcal{D}|)$. Similarly, for the next POI prediction task, that operation is conducted over all POI samples (S_l), as well as the generated POI samples ($\widehat{S_l}$). Besides, in the explicit strategy, we calculate the attention scores for each POI sample (S_l or $\widehat{S_l}$) with $|\mathcal{Y}|$ category representations from all cities. Therefore, the complexity of the self-attention operation for the next POI prediction task in CATUS is $O(2n^2d|\mathcal{D}|(|\mathcal{Y}|+1))$. In CATUS+DS, both the original POI samples (S_l) in each city y_i and the generated POI samples ($\widetilde{S_l}$) are used for fine-tuning, leading to the time complexity of $O(2n^2d|\mathcal{D}_i|)$.

To summarize, the total time complexity is $O(n|\mathcal{D}|+\rho n|\mathcal{D}|+n^2d|\mathcal{D}|+2n^2d|\mathcal{D}|(|\mathcal{Y}|+1))$ and $O(|\mathcal{L}|^2+\rho n|\mathcal{D}_i|+2n^2d|\mathcal{D}_i|)$ for CATUS at the pre-training stage and CATUS+DS at the fine-tuning stage, respectively.

4 EXPERIMENTS

In this section, we conduct a series of experiments to verify the effectiveness of our proposed CATUS.

4.1 Experiment Setup

4.1.1 Datasets. To qualitatively evaluate CATUS and CATUS+DS, we conduct experiments on two widely used public datasets collected from worldwide cities: Foursquare and Gowalla[5]. Foursquare contains users' check-in records from April 2012 to September 2013 while Gowalla contains records from February 2009 to October 2010. Each check-in record is associated with the user ID, POI ID and check-in time. Both datasets provide category information[6] and GPS coordinate information. To reduce the computational cost, we preprocess each large dataset by retaining two most active cities and their check-in records. Specifically, we select *Austin* and *San Francisco* in Gowalla, and *Tokyo* and *New York* in Foursquare. Furthermore, we only keep the most recently visited 50 check-ins for each user in Foursquare to avoid data redundancy. The statistics of preprocessed datasets are described in Table 2.

For each user, we split the corresponding check-in sequence into training, validation, and test parts, following Luo et al. [29].

4.1.2 Evaluation Metrics. To assess the next POI recommendation performance, we employ two widely used evaluation metrics: Hit Rate at Rank K (HR@K) and Normalized Discounted Cumulative Gain at Rank K (NDCG@K) [21, 23], where K is from $\{5, 10\}$. HR@K measures the fraction of the positive POI being included in the top K recommendation list. NDCG@K measures the quality of result recommendation list and assigns a higher weight to a higher rank position.

4.1.3 Pre-traing baseline methods. We compare our pre-training model with three state-of-the-art pre-training approaches for sequential POI recommendation:

[5]http://snap.stanford.edu/data/loc-gowalla.html
[6]The category information of Gowalla is provided by Yang et al. [51]

Table 2. Dataset Statistics.

Dataset	#Category	City	#User	#POI	#Check-in
Gowalla	387	Austin	8,276	21,352	332,680
		San Francisco	8,599	35,003	306,883
Foursquare	250	Tokyo	2,293	28,624	114,650
		New York	1,083	18,576	54,150

- **CLUE** [5] is a sequential behavior pre-training method. It employs sequence-level contrastive learning on two randomly augmented views of the same sequential behaviors, so as to learn effective user representations.
- **TALE** [42] is a time-aware location embedding pre-training method for next POI prediction. It is based on the CBOW framework and constructs Huffman tree with temporal information.
- **CTLE** [26] is a context and time aware location embedding pre-training method for next POI prediction. It is based on the BERT framework and introduces two self-supervised tasks: masked POI prediction and masked hour prediction.

Among them, CLUE is a pre-training method for the online recommendation which employs the prevalent contrastive learning technique and produces marvelous results over several pioneering baselines like S3Rec [62], thus is used as an indispensable baseline. TALE and CTLE are pre-training methods for next POI recommendation. To fully investigate the impacts of our proposed pre-training method under the same condition, we further propose two variants for TALE and CTLE by changing their time context into category. **TALE-C** builds the Huffman tree with category information instead of temporal information and **CTLE-C** replaces the masked hour prediction task with the masked category prediction task.

4.1.4 Downstream baseline methods. To examine the generalization ability for different downstream models, we apply CATUS over five representative sequential POI recommendation models:

- **STRNN** [27] is a variant of RNN. It introduces time-specific and distance-specific transition matrices into RNN for modeling temporal and spatial contexts.
- **STLSTM** [19] is a variant of LSTM. It introduces temporal and spatial factors to guide the learning process of LSTM gate mechanism.
- **SASRec** [17] is a self-attention based method. It directly utilizes the plausible transformer for the sequential behavior modeling. Although SASRec is not designed for the next POI recommendation, it achieves competitive performance.
- **STAN** [29] is a self-attention based sequential POI recommendation method. It considers spatiotemporal correlations of all the check-ins with a bi-attention architecture.
- **Graph-FlashBack** [34] is a state-of-the-art graph-based sequential POI recommendation method. It exploits relations among POIs and users with a constructed spatial-temporal knowledge graph.

4.1.5 Implementation Details. At the pre-training stage, we pre-train all models on the large dataset across cities. For pre-training baselines TALE and CTLE, we utilize the source codes provided by the authors. We implement TALE-C, CTLE-C, CLUE, and CATUS with Pytorch. For all pre-training methods, we set embedding size to 64 for a fair comparison. The other hyper-parameters of pre-training baselines are set according to the recommendation in their original papers. For CATUS, the hyperparameters ρ and λ are fixed to 0.3 and 0.5. The numbers of

Table 3. Performance comparison on four sub-datasets. *DM* denotes the downstream model for fine-tuning and *PM* denotes the pre-training model. *None* refers to training the downstream models with randomly initialized parameters. For each downstream model, the best scores are highlighted in bold and the best baseline scores are marked by underline. $*$ means the statistical significant improvement compared to the best baseline results ($p < 0.01$).

DM	PM	Gowalla								Foursquare							
		Austin				San Francisco				Tokyo				New York			
		HR@K		NDCG@K		HR@K		NDCG@K		HR@K		NDCG@K		HR@K		NDCG@K	
		K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10
STRNN	None	11.39	15.07	8.27	9.45	13.65	17.61	10.14	11.41	20.10	23.59	15.34	16.45	16.53	18.56	12.87	13.54
	CLUE	12.63	16.66	9.40	10.70	14.72	18.64	11.36	12.63	21.02	24.29	16.80	17.87	19.48	21.88	15.92	16.70
	TALE	12.94	17.21	9.40	10.78	16.95	20.98	13.02	14.32	20.93	24.73	16.43	17.65	20.87	24.10	15.88	16.93
	CTLE	13.77	18.32	10.21	11.68	18.61	22.61	14.36	15.65	29.31	35.19	22.53	24.42	30.29	34.72	23.63	25.06
	TALE-C	12.50	16.42	9.11	10.37	16.34	20.34	12.61	13.90	21.37	25.38	16.76	18.05	17.73	21.14	14.65	13.52
	CTLE-C	14.13	18.77	10.44	11.93	18.34	22.80	14.08	15.52	29.61	35.50	22.55	24.45	30.66	34.26	23.84	25.03
	CATUS	15.84*	20.42*	11.81*	13.31*	**19.88***	24.04*	**15.17***	**16.51***	30.31*	35.80	22.94	24.72	33.98*	38.23*	26.11*	27.51*
	CATUS+DS	**16.42***	**21.33***	**12.21***	**13.79***	19.31*	**24.13***	14.80	16.35*	**31.71***	**38.42***	**24.22***	**26.40***	**34.53***	**39.70***	**27.11***	**28.78***
STLSTM	None	13.91	17.61	10.41	11.60	19.62	23.48	15.64	16.89	27.69	31.40	21.42	22.62	29.09	32.87	23.19	24.43
	CLUE	13.85	17.84	10.47	11.76	15.28	19.12	12.06	13.31	22.76	25.64	17.90	18.84	27.70	30.10	21.39	22.19
	TALE	13.85	18.13	10.36	11.73	18.69	23.17	14.70	16.14	23.86	28.39	18.29	19.76	24.56	27.15	19.57	20.40
	CTLE	14.94	18.99	10.99	12.29	20.27	24.24	15.74	17.03	31.36	36.42	23.83	25.48	33.98	37.67	26.51	27.72
	TALE-C	14.59	18.98	10.82	12.23	18.72	22.57	14.42	15.67	24.90	29.18	19.15	20.54	20.78	23.45	16.12	16.99
	CTLE-C	14.96	19.16	11.24	12.59	20.39	24.70	15.80	17.18	31.23	37.29	24.01	25.97	33.80	39.06	26.22	27.94
	CATUS	15.87*	20.29*	11.80*	13.22*	20.41	24.83	15.94	17.38	31.97*	37.72*	24.33	26.21	35.49*	40.54*	27.50*	29.13*
	CATUS+DS	**16.55***	**21.42***	**12.30***	**13.88***	**20.99***	**25.56***	**16.29***	**17.76***	**32.75***	**38.64***	**25.12***	**27.04***	**35.73***	**41.18***	**27.62***	**29.40***
STAN	None	11.14	14.65	8.19	9.32	17.19	20.23	14.45	15.43	17.92	22.59	12.20	13.71	23.55	29.36	17.03	18.91
	CLUE	13.51	17.44	9.86	11.14	19.86	23.46	15.67	16.83	22.02	28.13	15.65	17.62	24.47	29.46	17.79	19.38
	TALE	13.22	17.16	9.92	11.17	18.02	20.89	14.91	15.82	22.11	28.48	16.09	18.17	23.82	29.73	17.70	19.60
	CTLE	9.17	12.81	6.94	8.11	14.79	17.71	11.87	12.82	16.96	20.41	13.08	14.20	13.67	16.81	10.39	11.38
	TALE-C	14.26	17.85	11.18	12.33	18.94	21.83	15.79	16.73	21.81	27.65	15.89	17.77	22.62	26.87	16.27	17.63
	CTLE-C	10.78	14.55	7.84	9.05	16.34	19.53	12.98	14.01	15.74	19.32	11.86	13.01	12.83	14.77	9.57	10.20
	CATUS	16.46*	21.14*	12.08*	13.59*	20.97*	25.68*	16.14	17.65*	30.14*	37.51*	21.74*	24.14*	32.87*	39.06*	24.23*	26.21*
	CATUS+DS	**16.53***	**21.58***	**12.28***	**13.91***	**22.75***	**27.56***	**17.19***	**18.75***	**31.14***	**38.20***	**23.40***	**25.68***	**33.24***	**39.24***	**24.82***	**26.76***
SASRec	None	15.20	20.32	10.71	12.37	20.39	24.57	14.66	16.01	23.59	29.39	17.26	19.15	25.30	29.18	18.65	19.90
	CLUE	15.66	20.54	11.33	12.90	20.80	25.04	15.79	17.16	25.08	30.96	18.13	20.01	27.52	31.58	20.38	21.73
	TALE	14.41	18.22	10.84	12.07	19.97	23.19	16.21	17.26	20.54	25.95	15.32	17.07	19.85	24.47	14.91	16.41
	CTLE	9.80	13.35	8.25	7.10	14.12	15.44	12.76	13.20	16.27	20.85	11.90	13.38	9.23	12.10	6.92	7.84
	TALE-C	15.14	18.99	10.90	12.15	19.91	23.15	16.03	17.07	20.15	25.34	15.05	16.73	19.21	24.93	13.58	15.45
	CTLE-C	10.35	14.28	7.65	8.92	15.06	18.53	11.80	12.92	14.52	18.84	10.95	12.37	9.79	12.74	7.25	8.20
	CATUS	16.92*	22.00*	12.52*	14.16*	22.66*	27.60*	17.24*	18.85*	31.28*	38.64*	22.67*	25.03*	35.73*	41.33*	26.31*	28.11*
	CATUS+DS	**17.20***	**22.53***	**12.80***	**14.52***	**23.72***	**28.79***	**18.17***	**19.81***	**33.14***	**41.13***	**24.80***	**27.38***	**37.03***	**42.47***	**27.56***	**29.28***
Graph-FlashBack	None	16.28	21.09	11.64	13.19	20.41	25.07	15.42	16.94	34.54	41.39	26.39	28.61	36.01	41.27	27.67	29.40
	CLUE	14.73	19.25	10.83	12.29	17.53	22.47	13.20	14.81	28.83	34.06	22.25	23.94	27.52	32.32	20.50	22.08
	TALE	16.39	21.67	12.09	13.79	20.57	25.56	15.87	17.51	35.36	41.91	27.23	29.37	36.57	41.78	28.20	29.90
	CTLE	16.10	20.92	11.65	13.20	19.86	24.39	15.55	17.02	33.84	40.08	25.94	27.97	35.55	41.37	27.50	29.38
	TALE-C	16.37	21.42	12.10	13.73	21.38	26.16	16.22	17.78	35.30	42.04	27.18	29.38	36.66	41.57	**28.50**	30.10
	CTLE-C	16.28	21.18	11.80	13.38	20.75	25.04	15.84	17.22	33.54	39.73	26.13	28.14	35.00	40.44	27.12	28.88
	CATUS	16.91*	22.16*	12.58*	14.28*	21.59	26.37	16.51	18.06	35.88	**42.77**	**27.36**	**29.60**	**36.73**	42.04	28.29	30.03
	CATUS+DS	**17.40***	**22.87***	**13.08***	**14.85***	**22.05***	**27.41***	**16.80***	**18.54***	**35.99**	42.76	27.30	29.50	36.59	**42.10**	28.40	**30.21**

attention blocks and heads are set to 2 respectively. We optimize the parameters by using the Adam optimizer with a learning rate of 0.0001 and a batch size of 256.

At the fine-tuning stage, for STRNN, STLSTM, and SASRec, we implement them with Pytorch. For STAN and Graph-FlashBack, we utilize the source codes provided by the authors. Here, all downstream models share the same learning rate of 0.0001. According to the validation performance, the batch sizes are set to 256 for SASRec, STAN, and Graph-FlashBack, while 128 for STRNN and STLSTM. Similarly, the hyper-parameter α is set to 0.5

for CATUS+DS. Our experiments were run on a machine equipped with a 3.70 GHz CPU, 64 GB RAM, and an NVIDIA 2080Ti GPU.

4.2 Performance Comparison

The experimental results on four city sub-datasets are summarized in Table 3. Here, we make the following observations:

Our proposed CATUS consistently improves the performance of all downstream models on four city sub-datasets. In detail, CATUS achieves 8.1% and 7.1% improvements of NDCG@5 on Austin and San Francisco, respectively, compared to the best downstream model Graph-FlashBack. Similarly, CATUS provides 16.26% and 22.77% average improvements over STLSTM on Tokyo and New York, respectively. Moreover, CATUS outperforms pre-training alternatives significantly in most cases. The average performance gains of CATUS are 8.43%, 4.4%, 12.57%, and 15.55% on Austin, San Francisco, Tokyo, and New York, respectively, against the best pre-training baseline. These results demonstrate the superiority of CATUS on exploiting universal transition-knowledge across cities.

CATUS+DS further enhances CATUS in most cases. This verifies the effectiveness of the distance-oriented sampler for fine-tuning. The generated POI sequences explicitly incorporate valuable spatial information, and oblige the downstream model to smooth the representations of the neighbor POIs. In general, the distance-oriented sampler is effective to fine-tune the representations into the local context in each city.

We observe that existing pre-training baselines may introduce some negative impact in some cases. For example, SASRec and STAN experience an obvious performance degradation when CTLE is chosen for pre-training. The possible reason is that CTLE uses point-wise loss for masked POI/hour prediction while SASRec and STAN uses pair-wise loss, which leads to an incompatibility problem between CTLE and these two downstream models. Furthermore, TALE degrades the performance of SASRec and STLSTM, while CLUE degrades the performance of STLSTM and Graph-FlashBack. An explanation could be that TALE and CLUE are deficient in modeling sequential context, thus hurt the optimization of these downstream models. On the contrary, CATUS substantially improves the prediction accuracy of all downstream models regardless of different model designs, further guaranteeing the strong generalization ability.

TALE and TALE-C help Graph-FlashBack achieve a comparable performance with our CATUS. This might be due to fact that TALE and TALE-C incorporate temporal or category influence into the building process of Huffman tree, which explicitly connects POIs from different cities. Note that the graph-based model Graph-FlashBack also connects POIs explicitly. Thus TALE and TALE-C are compatible with Graph-FlashBack and produce a better performance.

At last, compared to CTLE and TALE, CTLE-C and TALE-C with category information cannot consistently produce superior performance. This suggests that the shared set of categories across cities requires sophisticated exploitation, *e.g.*, transferring knowledge from universal category to specific POI. A simple strategy like parameter-sharing cannot take advantage of different data across cities.

4.3 Ablation Study

Impact of Model Component. Here, we further investigate the impact of each design choice in CATUS with five variants:

- *w/o CS*: it removes the category-transition oriented sampler;
- *w/o IS*: it removes the implicit transfer strategy (without parameter sharing);
- *w/o ES*: it removes the explicit transfer strategy (without weighing POI vectors with category relevancy);
- *w/o CP*: it removes the next category prediction task;

Table 4. Ablation analysis on all sub-datasets with SASRec as the downstream model.

DM	PM	Gowalla								Foursquare							
		Austin				San Francisco				Tokyo				New York			
		HR@K		NDCG@K		HR@K		NDCG@K		HR@K		NDCG@K		HR@K		NDCG@K	
		K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10
SASRec	None	15.20	20.32	10.71	12.37	20.39	24.57	14.66	16.01	23.59	29.39	17.26	19.15	25.30	29.18	18.65	19.90
	w/o CS	16.25	21.09	11.86	13.42	21.39	26.50	16.07	17.73	28.52	34.76	20.11	22.12	28.44	34.72	20.24	22.26
	w/o IS	16.28	20.84	12.05	13.52	21.27	26.49	16.08	17.78	29.79	37.64	21.66	24.20	34.53	40.72	26.04	28.05
	w/o ES	16.37	21.63	12.11	13.80	22.11	27.22	16.72	18.38	30.89	38.29	22.55	24.94	35.21	40.75	**26.31**	28.10
	w/o CP	16.44	21.42	12.03	13.64	21.83	26.96	16.41	18.07	29.00	35.98	20.67	22.93	35.09	40.26	25.73	27.42
	w/o C	16.13	20.49	11.47	12.89	20.42	25.46	14.91	16.54	26.65	32.88	18.95	20.97	28.25	33.33	20.49	22.13
	CATUS	**16.92**	**22.00**	**12.52**	**14.16**	**22.66**	**27.60**	**17.24**	**18.85**	**31.28**	**38.64**	**22.67**	**25.03**	**35.73**	**41.33**	**26.31**	**28.11**

- *w/o C*: it removes all category-related mechanisms, including the category-transition oriented sampler, the implicit and explicit transfer strategy, and the next category prediction task.

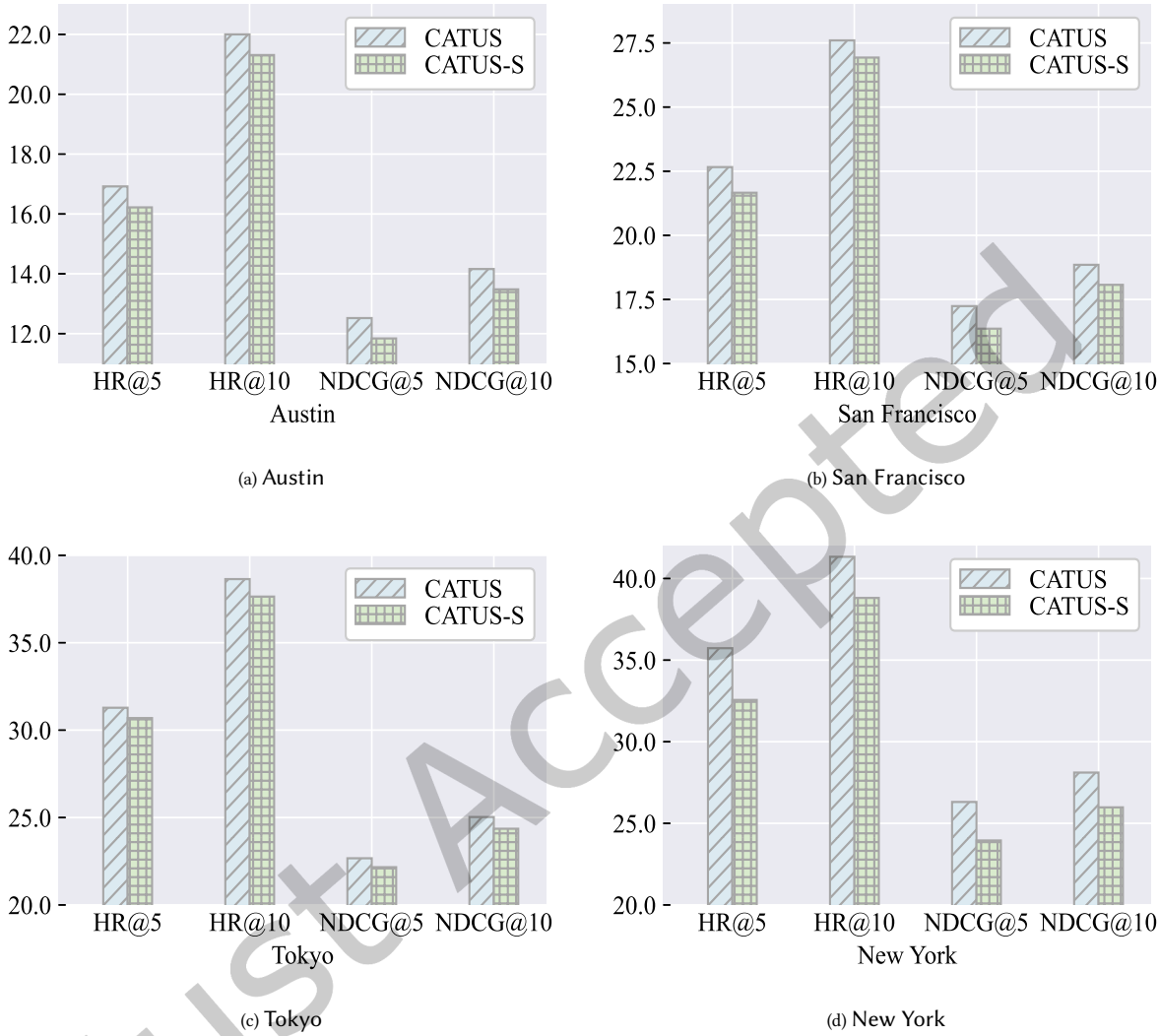
As SASRec is a succinct self-attention based model and shows competitive performance on most sub-datasets, we choose it as the downstream model and present the fine-tuning results on all city sub-datasets in Table 4.

The results of all variants are inferior to CATUS, which comfirms the positive effect of each part. By removing category-transition oriented sampler, *w/o CS* experiences a severer performance drop than *w/o IS* and *w/o ES*, suggesting that injecting universal knowledge into POI sequence data on the data level is more important. Similarly, comparing *w/o IS* and *w/o ES*, the positive impact of the implicit transfer strategy is also obvious. This phenomenon is reasonable since the implicit strategy of sharing parameters connects POI and category representations, which makes it feasible to encode POI sequence by category relevancy in the explicit transfer strategy. As for *w/o CP*, removing the next category prediction task weakens the capability of exploiting universal transition-knowledge. Lastly, *w/o C* delivers a very obvious performance degradation on all datasets. This implies that the category information is of great importance for addressing the lack of overlapping POIs and users in pre-training across cities. We also find an interesting phenomenon that *w/o C* still brings benefit to downstream model SASRec. One possible reason is that the shared sequential encoder projects POI embeddings from all cities into the same latent space, potentially modeling the latent correlations among different cities.

Impact of Pre-training Across Two Cities. We additionally adopt CATUS on each single city sub-dataset (namely, CATUS-S), and show results in Figure 7. Here, we can observe that CATUS outperforms CATUS-S on the city sub-dataset. This is consistent with our expectation that the universal transition-knowledge across cities is beneficial for next POI recommendation.

Impact of Pre-training Across More Cities. To verify whether our model works well or not on datasets consisting of more than two cities, we conduct an experiment on a three-city dataset by introducing a new city Dallas into the original two-city Gowalla dataset. Note that Dallas is the third active American city in Gowalla dataset (Austin is the first and San Francisco is the second). It contains 119,116 check-ins by 4,828 users and 15,320 POIs.

For simplicity, we choose SASRec as the downstream model, due to the same reason mentioned before that SASRec is a succinct self-attention based model and shows competitive performance on most sub-datasets. We rerun all pre-training methods on the newly constructed three-city dataset and present the fine-tuning results in Table 5 below:



(a) Austin (b) San Francisco

(c) Tokyo (d) New York

Fig. 7. Comparison between CATUS and CATUS-S.

From the results, we find that CATUS and CATUS+DS still outperform all pre-training baseline methods and achieve the best results for all three cities with SASRec as the downstream model. This proves that our proposed model works well on datasets consisting of more than two cities. Furthermore, we find that the scores on Austin and San Francisco cities in Table 5 are very close to those scores by pre-training without Dallas in Table 3. This might be due to the fact that Dallas has much less check-in records than Austin and San Francisco, thus providing limited knowledge for Austin and San Francisco.

Table 5. Results on a three-city dataset with SASRec as the downstream model.

PM	Gowalla											
	Austin				San Francisco				Dallas			
	HR@K		NDCG@K		HR@K		NDCG@K		HR@K		NDCG@K	
	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10	K=5	K=10
None	15.20	20.32	10.71	12.37	20.39	24.57	14.66	16.01	16.84	21.04	12.22	13.59
CLUE	15.21	19.91	11.15	12.66	19.31	23.45	14.25	15.58	17.45	21.62	12.87	14.23
TALE	15.06	19.24	11.43	12.78	19.13	22.64	15.24	16.38	16.20	19.38	12.22	13.24
CTLE	10.96	14.94	8.06	9.34	15.95	19.65	12.28	13.47	12.05	15.86	9.17	10.39
TALE-C	14.78	18.73	10.56	11.84	19.76	22.87	15.68	16.70	15.60	19.01	11.72	12.82
CTLE-C	10.51	14.58	7.69	9.00	17.27	20.89	13.52	14.70	11.31	15.09	8.52	9.75
CATUS	16.92	22.09	12.53	14.20	22.72	27.46	17.05	18.58	19.72	24.74	14.61	16.21
CATUS+DS	**17.14**	**22.66**	**12.56**	**14.34**	**23.31**	**28.64**	**17.37**	**19.09**	**20.54**	**25.24**	**15.69**	**17.21**

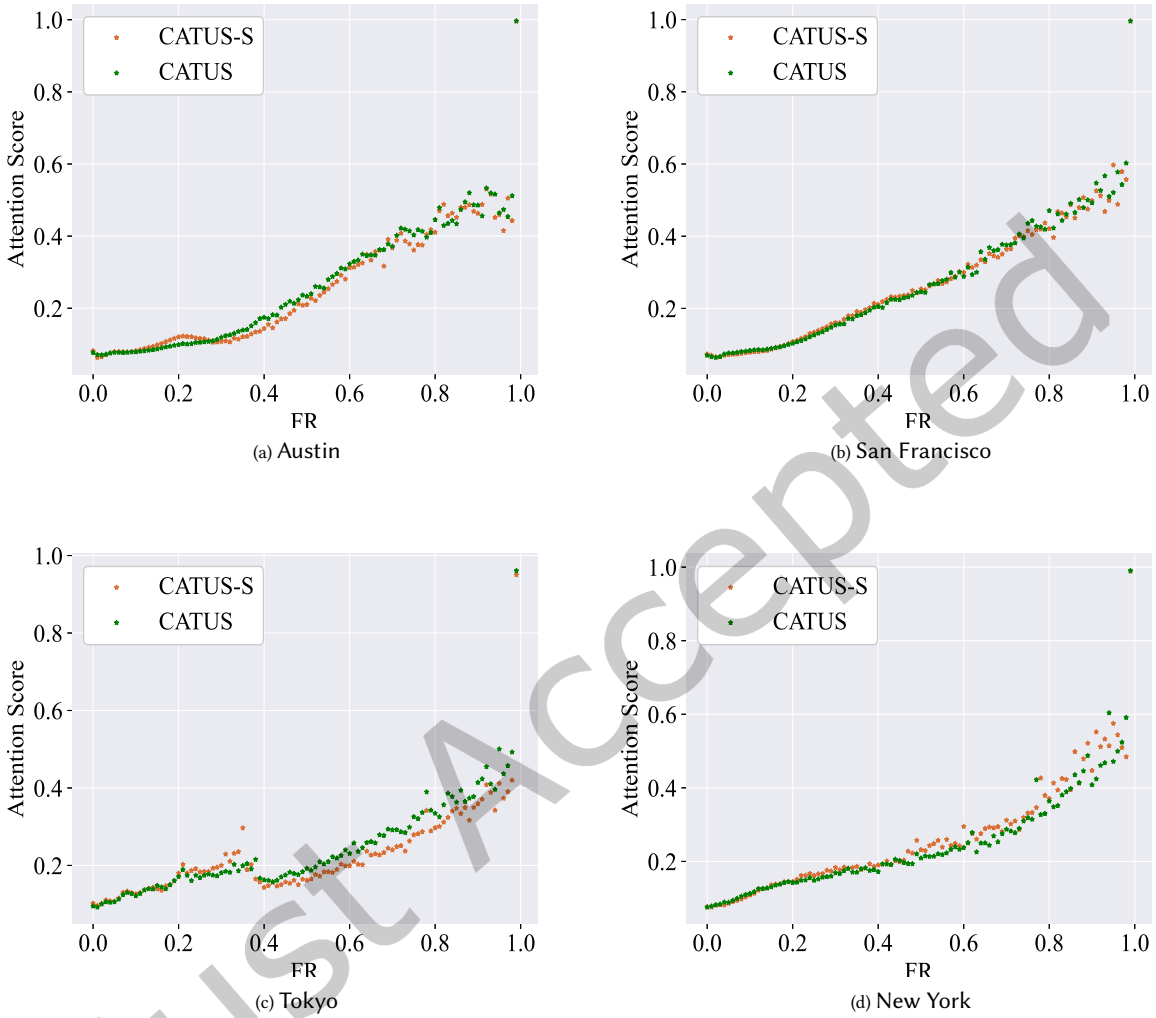
4.4 Detailed Analysis

4.4.1 In-depth Analysis. As aforementioned in Section 1, the combined large dataset across cities brings two advantages to alleviate the sparsity problem: *knowledge transfer* and *knowledge enhancement*. Knowledge transfer refers to transferring a frequent pattern in one city to another city, while knowledge enhancement refers to enhancing the modeling of a sparse pattern in multiple cities. Here we conduct an in-depth analysis to verify whether our proposed CATUS achieves the desired goals, by investigating the attention score and relative sparsity of a category transition. The attention score is from the final layer of CATUS and CATUS-S after fine-tuning with SASRec, while the relative sparsity is from the dataset statistics.

Given a historical sequence S_h, there is a set of corresponding visited categories C_h and a next visited category c_t. In the final layer of a pre-trained model, each category transition $c_h \rightarrow c_t, c_h \in C_h$ can obtain a category attention score. It is calculated by adding up attention scores of POIs belonging to c_h in S_h. To assess the relative sparsity, we introduce a frequency ratio (FR) for each category transition in S_h. In detail, we first calculate the occurrence frequency $f_{c_h \rightarrow c_t}$ of $c_h \rightarrow c_t, c_h \in C_h$ in the dataset. Then, we formally define FR for $c_h \rightarrow c_t$ in S_h as $FR_{c_h \rightarrow c_t} = f_{c_h \rightarrow c_t} / \sum_{c_i \rightarrow c_t, c_i \in C_h} f_{c_i \rightarrow c_t}$, where a larger $FR_{c_h \rightarrow c_t}$ means that $c_h \rightarrow c_t$ is denser in the dataset.

We analyze the relationship between the attention score and FR on the training dataset, and show the result in Figure 8. We find that, for both CATUS and CATUS-S, the attention score gradually increases when FR grows. This shows that the pre-trained model favors relatively denser transition patterns in a sequence. Based on this observation, we can assume a case where a lower FR transition in the sub-dataset becomes a higher FR one in the combined large dataset, and would be assigned a greater attention weight by CATUS. Therefore, it is reasonable to conjecture that CATUS has the ability to transfer or enhance knowledge across cities.

To have a deep look at CATUS, we first take two examples from Tokyo and New York to illustrate knowledge transfer ability of CATUS. Figure 9 shows these two knowledge transfer examples *a* and *b*. For both of them, CATUS accurately predicts the next target POI while CATUS-S makes mistakes. In Figure 9, we visualize the attention scores of each category transition assigned by CATUS-S and CATUS, as well as frequency ratios in the city sub-dataset and the combined dataset. We also show the target category of each sample in the caption. From Figure 9, It is clearly observed that the FRs of some category transitions reach a higher value after combining two sub-datasets, *e.g., Bar→Office* and *Food&Drink Shop → Train Station*. Consequently, CATUS pays more attention to these category transitions and makes the correct recommendation. This gives evidence that CATUS transfers knowledge among different cities.



(a) Austin (b) San Francisco

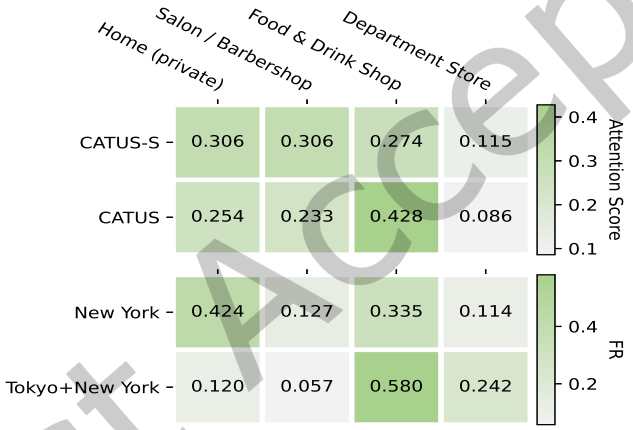
(c) Tokyo (d) New York

Fig. 8. Relationship between frequency ratio and attention score on four city sub-datasets.

We then take two examples from Tokyo and New York to illustrate knowledge enhancement ability of CATUS. Figure 10 shows these two knowledge enhancement examples *c* and *d*. Similarly, CATUS accurately predicts the next target POI for these two examples while CATUS-S makes mistakes. In Figure 10, CATUS apparently turns attention to the relatively sparser transitions with lower FR scores, such as *Food&Drink Shop* → *College Academic Building* in example *c* and *Subway* → *Home (private)* in example *d*. This suggests that CATUS achieves knowledge enhancement on the combined dataset.

In summary, CATUS indeed takes good advantage of pre-training across cities and facilitates alleviating both absolute sparsity and relative sparsity problems.

(a) The example a from Tokyo. The target category is *Office*.

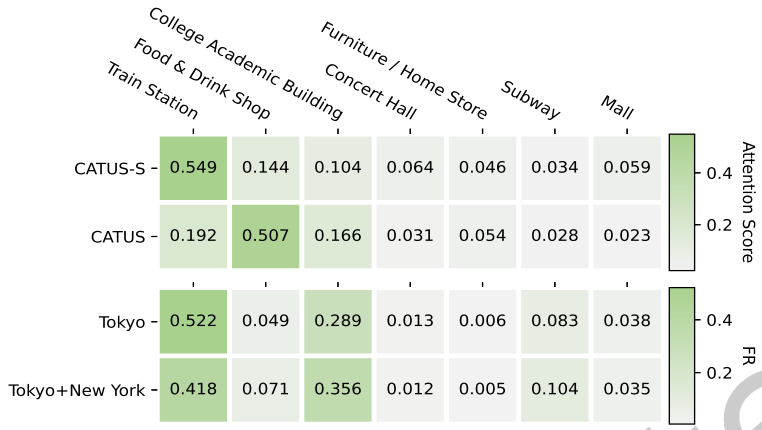


(b) The example b from New York. The target category is *Train Station*.

Fig. 9. The example for knowledge transfer.

4.4.2 Impacts of Hyperparameters. We further investigate the impacts of λ and α to CATUS and CATUS+DS, respectively. λ is for category-transition oriented sampler in CATUS while α for distance oriented sampler in CATUS+DS. The larger the value, the less impact of augmented POI samples generated. We present the NDCG@10 results in Figure 11. We find that the recommendation performance reaches optimal when λ and α are around 0.5, indicating that an appropriate balance between observed and augmented samples contributes to the prediction accuracy. This confirms that both types of augmented POI sequences play critical roles.

4.4.3 Computational Cost Analysis. To show the efficiency of our proposed method, we compare the computational cost with baselines w.r.t. pre-training and fine-tuning stages. For the pre-training stage, we show



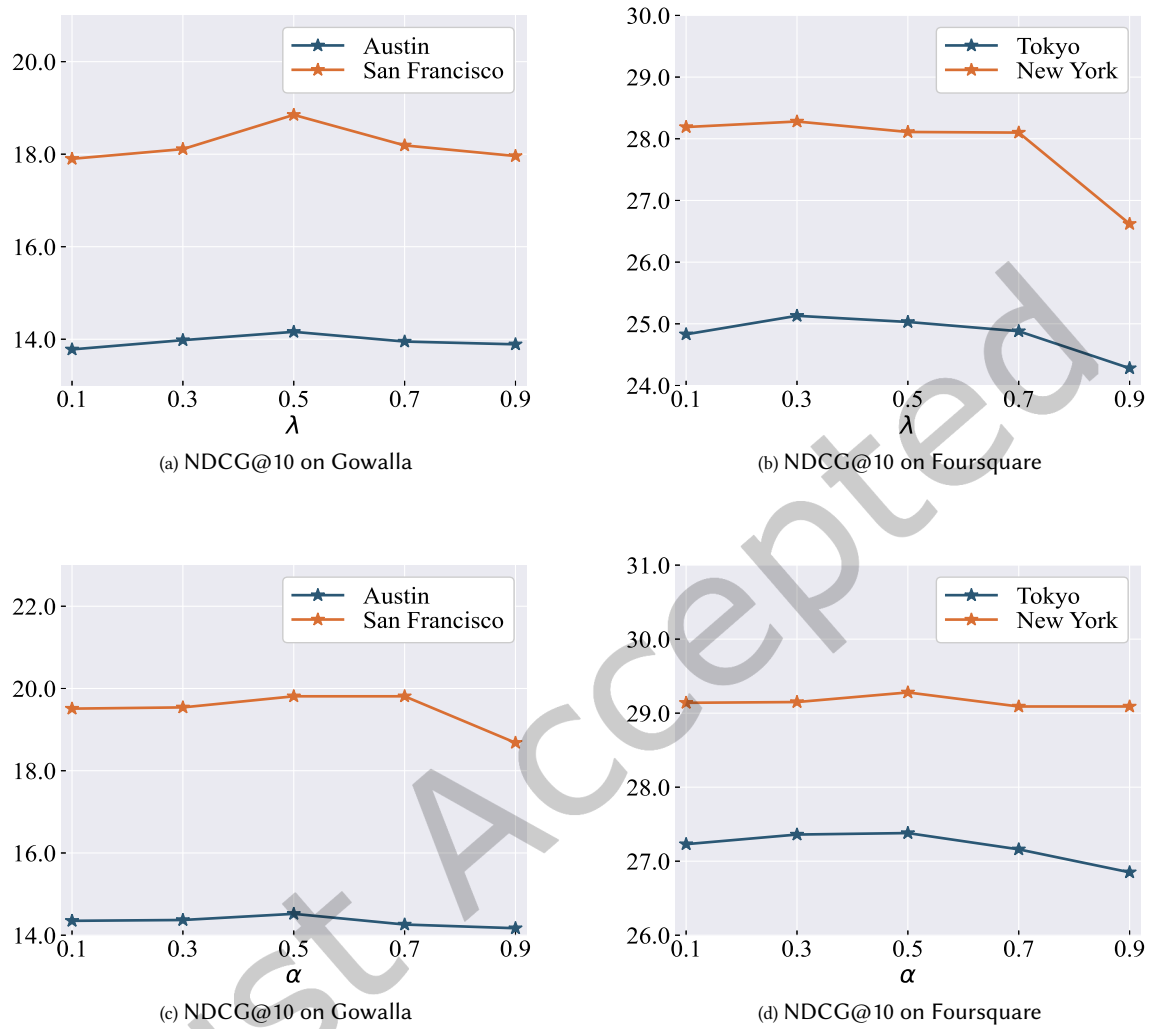
(c) The example c from Tokyo. The target category is *College Academic Building*.

(d) The example d from New York. The target category is *Home (private)*.

Fig. 10. The example for knowledge enhancement.

the runtime on the two-city Gowalla dataset, while for the fine-tuning stage, we show the runtime on Austin sub-dataset with SASRec as the downstream model. The results are shown in Table 6.

We can see that, for the pre-training stage, CLUE is the most efficient method since it only adopts the contrastive learning task without any prediction task. Our method CATUS is computationally comparable with CTLE and CTLE-C. This is because that CATUS is a transformer-based model with two prediction tasks, which is very similar to CTLE and CTLE-C. TALE and TALE-C also require prediction tasks, but they don't need any sequential encoder, making them moderately efficient. For the fine-tuning stage, our method is much efficient compared to other baselines, because CATUS provides better representations for SASRec and makes SASRec converge faster. CATUS+DS takes more fine-tuning time owing to the augmented POI sequences which requires more sequence encoding time. Overall, we can make a conclusion that the computational cost of our model is acceptable.



(a) NDCG@10 on Gowalla

(b) NDCG@10 on Foursquare

(c) NDCG@10 on Gowalla

(d) NDCG@10 on Foursquare

Fig. 11. The impacts of λ and α to CATUS and CATUS+DS with SASRec as the downstream model on four sub-datasets. Note that CATUS+DS is conducted with λ set to 0.5 at the pre-training stage.

5 CONCLUSION

In this paper, we advocate a new problem of pre-training across different cities for next POI recommendation, and we develop a new model CATUS which utilizes category information to fully exploit the universal transition-knowledge across cities. In particular, we first propose next category and POI prediction tasks to learn the universal transition-knowledge across cities. Then, to pass the knowledge from the universal category into specific POI, we develop a category-transition oriented sampler, and an implicit and explicit strategy to enhance this process. Finally, we introduce a distance oriented sampler to guide the fine-tuning. Our experimental results on two combined large datasets demonstrate that CATUS improves the performance of downstream models and

Table 6. Analysis on computational cost.

PM	Pre-training Runtime	Fine-tuning Runtime
CLUE	240.74s	3232.96s
TALE	4098.01s	3283.56s
CTLE	15587.52s	3841.93s
TALE-C	3182.18s	3244.22s
CTLE-C	15677.88s	3917.22s
CATUS	15163.34s	1902.46s
CATUS+DS	-	4654.35s

outperforms state-of-the-art pre-training methods. Also, the distance oriented sampler further promotes the fine-tuning performance of CATUS.

We have proved that transition knowledge can be transferred across different cities. This can largely mitigate the sparsity problem in those cities with limited collected check-in data, and consequently improve the POI recommendation performance. Therefore, we believe that a positive effect of this work is to promote economic development of small cities by learning from large cities and recommending suitable POIs, e.g., stores and restaurants, to people living in small cities. However, a limitation of this work is that, when a new city comes, it is required to re-pretrain the large dataset for knowledge transfer, which is impractical. In the future, we would like to make this process more practical, and study how to learn from other cities while avoiding training over the combined large dataset.

ACKNOWLEDGMENT

This work has been supported in part by the NSFC Projects (62276193, 62032016, 61972291).

REFERENCES

[1] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation.. In *IJCAI*. 3301–3307.
[2] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. 2018. " Bridge" Enhanced Signed Directed Network Embedding. In *CIKM*. 773–782.
[3] Yudong Chen, Xin Wang, Miao Fan, Jizhou Huang, Shengwen Yang, and Wenwu Zhu. 2021. Curriculum meta-learning for next POI recommendation. In *KDD*. 2692–2702.
[4] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*. 2605–2611.
[5] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning Transferable User Representations with Sequential Behaviors via Contrastive Pre-training. In *ICDM*. 51–60.
[6] Qiang Cui, Chenrui Zhang, Yafeng Zhang, Jinpeng Wang, and Mingchen Cai. 2021. ST-PIL: Spatial-Temporal Periodic Interest Learning for Next Point-of-Interest Recommendation. In *CIKM*. 2960–2964.
[7] Yue Cui, Hao Sun, Yan Zhao, Hongzhi Yin, and Kai Zheng. 2021. Sequential-knowledge-aware next POI recommendation: A meta-learning approach. *TOIS* 40, 2 (2021), 1–22.
[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
[9] 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 *WWW*. 1459–1468.
[10] 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 *IJCAI*. 2069–2075.
[11] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An attentional recurrent neural network for personalized next location recommendation. In *AAAI*. 83–90.

[12] Bowen Hao, Hongzhi Yin, Jing Zhang, Cuiping Li, and Hong Chen. 2021. A Multi-Strategy based Pre-Training Method for Cold-Start Recommendation. *arXiv preprint arXiv:2112.02275* (2021).
[13] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-Training Graph Neural Networks for Cold-Start Users and Items Representation. In *WSDM*. 265–273.
[14] Jing He, Xin Li, and Lejian Liao. 2017. Category-aware Next Point-of-Interest Recommendation via Listwise Bayesian Personalized Ranking.. In *IJCAI*. 1837–1843.
[15] Jing He, Xin Li, Lejian Liao, Dandan Song, and William Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*. 137–143.
[16] Liwei Huang, Yutao Ma, Yanbo Liu, and Keqing He. 2020. DAN-SNR: A deep attentive network for social-aware next point-of-interest recommendation. *ACM Transactions on Internet Technology* 21, 1 (2020), 1–27.
[17] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. 197–206.
[18] Minseok Kim, Hwanjun Song, Doyoung Kim, Kijung Shin, and Jae-Gil Lee. 2021. PREMERE: Meta-Reweighting via Self-Ensembling for Point-of-Interest Recommendation. In *AAAI*. 4164–4171.
[19] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A Hierarchical Spatial-Temporal Long-Short Term Memory Network for Location Prediction.. In *IJCAI*. 2341–2347.
[20] Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *ICDM*. 1110–1115.
[21] Yang Li, Tong Chen, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. In *IJCAI*. 1491–1497.
[22] Zeyu Li, Wei Cheng, Haiqi Xiao, Wenchao Yu, Haifeng Chen, and Wei Wang. 2021. You Are What and Where You Are: Graph Enhanced Attention Network for Explainable POI Recommendation. In *CIKM*. 3945–3954.
[23] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-aware sequential location recommendation. In *KDD*. 2009–2019.
[24] 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. (2022), 1133–1143.
[25] 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 *CIKM*. 845–854.
[26] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2021. Pre-training Context and Time Aware Location Embeddings from Spatial-Temporal Trajectories for User Next Location Prediction. In *AAAI*. 4241–4248.
[27] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*. 194–200.
[28] Yong Liu, Susen Yang, Chenyi Lei, Guoxin Wang, Haihong Tang, Juyong Zhang, Aixin Sun, and Chunyan Miao. 2021. Pre-training Graph Transformer with Multimodal Side Information for Recommendation. In *MM*. 2853–2861.
[29] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *WWW*. 2177–2185.
[30] Xuan Ma, Tieyun Qian, Yile Liang, Ke Sun, Hang Yun, and Mi Zhang. 2022. Enhancing Graph Convolution Network for Novel Recommendation. In *DASFAA*. Springer, 69–84.
[31] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2018. A contextual attention recurrent architecture for context-aware venue recommendation. In *SIGIR*. 555–564.
[32] Tieyun Qian, Yile Liang, Qing Li, Xuan Ma, Ke Sun, and Zhiyong Peng. 2022. Intent disentanglement and feature self-supervision for novel recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[33] Tieyun Qian, Yile Liang, Qing Li, and Hui Xiong. 2020. Attribute graph neural networks for strict cold start recommendation. *TKDE* (2020).
[34] Xuan Rao, Lisi Chen, Yong Liu, Shuo Shang, Bin Yao, and Peng Han. 2022. Graph-Flashback Network for Next Location Recommendation. In *KDD*. 1463–1471.
[35] Huimin Sun, Jiajie Xu, Kai Zheng, Pengpeng Zhao, Pingfu Chao, and Xiaofang Zhou. 2021. MFNP: A Meta-optimized Model for Few-shot Next POI Recommendation. In *IJCAI*. 3017–3023.
[36] Ke Sun and Tieyun Qian. 2018. Exploiting user and item attributes for sequential recommendation. In *ICONIP*. Springer, 370–380.
[37] 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. In *AAAI*. 214–221.
[38] Ke Sun, Tieyun Qian, Xu Chen, and Ming Zhong. 2021. Context-aware seq2seq translation model for sequential recommendation. *Information Sciences* 581 (2021), 60–72.
[39] Ke Sun, Tieyun Qian, Hongzhi Yin, Tong Chen, Yiqi Chen, and Ling Chen. 2019. What Can History Tell Us?. In *CIKM*. 1593–1602.
[40] Ke Sun, Tieyun Qian, Ming Zhong, and Xuhui Li. 2023. Towards more effective encoders in pre-training for sequential recommendation. *World Wide Web* (2023), 1–32.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.
[42] Huaiyu Wan, Yan Lin, Shengnan Guo, and Youfang Lin. 2021. Pre-training Time-Aware Location Embeddings from Spatial-Temporal Trajectories. *TKDE* (2021).
[43] Chen Wang, Yueqing Liang, Zhiwei Liu, Tao Zhang, and Philip S Yu. 2021. Pre-training Graph Neural Network for Cross Domain Recommendation. *arXiv preprint arXiv:2111.08268* (2021).
[44] Lingzhi Wang, Xingshan Zeng, Huang Hu, Kam-Fai Wong, and Daxin Jiang. 2021. Re-entry Prediction for Online Conversations via Self-Supervised Learning. In *Findings of EMNLP*. 2127–2137.
[45] Peng Wang, Jiang Xu, Chunyi Liu, Hao Feng, Zang Li, and Jieping Ye. 2020. Masked-field Pre-training for User Intent Prediction. In *CIKM*. 2789–2796.
[46] Xiaolin Wang, Guohao Sun, Xiu Fang, Jian Yang, and Shoujin Wang. 2022. Modeling Spatio-temporal Neighbourhood for Personalized Point-of-interest Recommendation. In *IJCAI*. 3530–3536.
[47] Zhaobo Wang, Yanmin Zhu, Haobing Liu, and Chunyang Wang. 2022. Learning Graph-based Disentangled Representations for Next POI Recommendation. In *SIGIR*. 1154–1163.
[48] Chuhan Wu, Fangzhao Wu, Tao Qi, Jianxun Lian, Yongfeng Huang, and Xing Xie. 2020. PTUM: Pre-training User Model from Unlabeled User Behaviors via Self-supervision. *arXiv preprint arXiv:2010.01494* (2020).
[49] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Xing Xie. 2021. UserBERT: Contrastive User Model Pre-training. *arXiv preprint arXiv:2109.01274* (2021).
[50] Chaojun Xiao, Ruobing Xie, Yuan Yao, Zhiyuan Liu, Maosong Sun, Xu Zhang, and Leyu Lin. 2021. UPRec: User-Aware Pre-training for Recommender Systems. *arXiv preprint arXiv:2102.10989* (2021).
[51] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *KDD*. 1245–1254.
[52] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *SIGIR*. 1144–1153.
[53] Di Yao, Chao Zhang, Jianhui Huang, and Jingping Bi. 2017. Serm: A recurrent model for next location prediction in semantic trajectories. In *CIKM*. 2411–2414.
[54] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*. 325–334.
[55] Fuqiang Yu, Lizhen Cui, Wei Guo, Xudong Lu, Qingzhong Li, and Hua Lu. 2020. A category-aware deep model for successive poi recommendation on sparse check-in data. In *WWW*. 1264–1274.
[56] Xu Yuan, Hongshen Chen, Yonghao Song, Xiaofang Zhao, Zhuoye Ding, Zhen He, and Bo Long. 2021. Improving Sequential Recommendation Consistency with Self-Supervised Imitation. In *IJCAI*. 3321–3327.
[57] Lu Zhang, Zhu Sun, Ziqing Wu, Jie Zhang, Yew Soon Ong, and Xinghua Qu. [n. d.]. Next Point-of-Interest Recommendation with Inferring Multi-step Future Preferences.
[58] Lu Zhang, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden, and Felix Klanner. 2020. An interactive multi-task learning framework for next POI recommendation with uncertain check-ins. In *IJCAI*. 3551–3557.
[59] Mingwei Zhang, Yang Yang, Rizwan Abbas, Ke Deng, Jianxin Li, and Bin Zhang. 2021. SNPR: A Serendipity-Oriented Next POI Recommendation Model. In *CIKM*. 2568–2577.
[60] 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. In *AAAI*. 5877–5884.
[61] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R Lyu, and Irwin King. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *AAAI*. 315–322.
[62] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*. 1893–1902.