International Journal of Innovative
Computing, Information and Control
Volume **20**, Number **5**, October **2024**

 ISSN 1349-4198
pp. **1277–1299**

PROXIMITY-AWARE SELF-ATTENTION-BASED SEQUENTIAL LOCATION RECOMMENDATION

Xuan Luo[1,*], Mingqing Huang[2], Rui Lv[2] and Hui Zhao[2]

[1]School of Digital Media
[2]School of Computer Science and Software Engineering
Shenzhen Institute of Information Technology
No. 2188, Longxiang Avenue, Longgang District, Shenzhen 518172, P. R. China
{ mqhuang; lvrui; zhaohui }@sziit.edu.cn
[*]Corresponding author: luoxuan@sziit.edu.cn

Received November 2023; revised March 2024

Abstract. *Sequential location recommendation plays a huge role in modern life, which can enhance user experience, bring more profit to businesses and assist in government administration. Although methods for location recommendation have evolved significantly thanks to the development of recommendation systems, there is still limited utilization of geographic information, along with the ongoing challenge of addressing data sparsity. In response, we introduce a Proximity-Aware based region representation for Sequential Recommendation (PASR for short), built upon the Self-Attention Network architecture. We tackle the sparsity issue through a novel loss function employing importance sampling, which emphasizes informative negative samples during optimization. Moreover, PASR enhances the integration of geographic information by employing an encoder based on self-attention to the hierarchical grid and proximity grid at each GPS point. To further leverage geographic information, we utilize the proximity-aware based negative sampling method to enhance the quality of negative training instances. We conducted evaluations using three real-world Location-Based Social Networking (LBSN) datasets, demonstrating that PASR surpasses State-of-the-Art (SOTA) sequential location recommendation solutions.*

Keywords: Sequential recommendation, GeoHash, Proximity region, Self-attention, Encoder

1. **Introduction.** In the era of rapid information technology advancement, the process of digitizing and sharing human mobility behaviors with friends has become significantly streamlined. These mobility patterns offer valuable insights into understanding and forecasting human movements [1, 2], thereby enhancing various aspects of daily life such as dining, transportation, and entertainment. However, the predictability of individual mobility remains a challenge [3, 4] due to data gaps and sparsity. To predict a personalized ranking of locations based on an individual's mobility history, sequential location recommendation assumes a pivotal role in enhancing the predictability of human movement across unfamiliar places. This is achieved by harnessing collective insights. Beyond its impact on mobility prediction, sequential location recommendation finds utility across a spectrum of applications, including route planning and location-targeted advertising.

Recently, the techniques employed for sequential location recommendation have witnessed a notable evolution, progressing from matrix factorization based solutions to the utilization of artificial neural networks, like RNN or CNN. To illustrate, the expansion of

DOI: 10.24507/ijicic.20.05.1277

Factorizing Personalized Markov Chains (FPMC) [5] was undertaken to address the challenge of sparse representation in modeling personalized location transitions [6, 7]. In the domain of metric learning, the introduction of Personalized Ranking Metric Embedding (PRME) was intended to characterize the individualized patterns of location transitions [8]. This concept was subsequently extended to encompass geographic influence by incorporating the product of travel distance and estimated transition probability. For capturing long-term relationship, the incorporation of RNNs, like GRU or LSTM, has been applied to integrate spatial-temporal patterns [2, 9, 10, 11, 12]. This was achieved by embedding parameters such as routing distance, and travel time. Moreover, the design of spatial-temporal gates to better utilize the spatial-temporal information has also been explored.

Among the prevailing methods, two noteworthy challenges remain inadequately addressed. Firstly, the effective utilization of geographic information continues to be a gap. It is widely acknowledged that the GPS coordinates of a location play a vital role in illustrating the distance between locations. Additionally, a user's historical mobility data often demonstrates a propensity for spatial clustering [13, 14]. Thus, the accurate encoding of GPS positions becomes indispensable. Secondly, these methods might grapple with the issue of sparsity. It is important to highlight that a user typically frequents a finite set of locations [15], leading to a mixture of negatively preferred and potentially positive locations within individual unvisited places [16]. Currently, the optimization in existing methods is achieved using either the BPR loss [17] or the BCE loss. This is done by comparing visited positions with randomly chosen samples from unvisited positions. However, each sample's level of informativeness varies, making it clear that treating all negative samples equally in these functions falls short of the optimal approach.

To overcome the above problems, we introduce a Proximity-Aware based region representation for Sequential Recommendation, referred to as PASR, built upon the foundation of the Self-Attention Network. PASR aims to enhance sequential location recommendation. As well as incorporating location embeddings, PASR introduces a novel geography encoder and a grid mapper to make full use of geography information. The geography encoder first utilizes the GeoHash to encode a geographic coordinate to a string of letters and digits. For instance, the GPS coordinates $(40.68925, -74.0445)$ of the Statue of Liberty are converted into a GeoHash string 'dr5r7p62n1' using the GeoHash algorithm, with a string length specified as 12. In Figure 1, we give the first three layers of GeoHash string w.r.t. the Statue of Liberty. Embedding the GeoHash string directly might seem straightforward, but it fails to capture complex spatial relationships. Note that the GeoHash strings of adjacent coordinates are similar and share some substrings. Therefore,

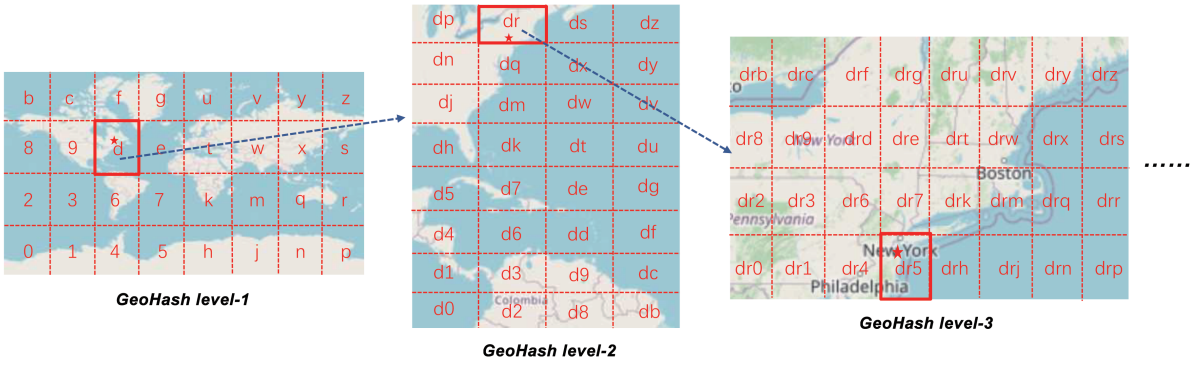

FIGURE 1. GeoHash example

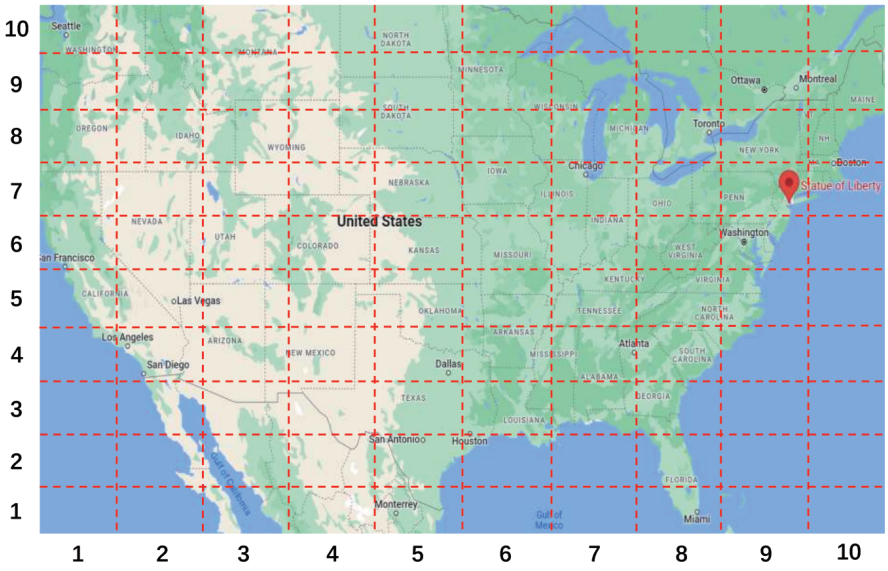

FIGURE 2. Grid partition

we leverage the self-attentive based encoder to encode the n-gram sequences of GeoHash strings. To enhance the proximity among positions, we propose to partition the region into grids. Concretely, the grid mapper divides the map into grids along latitude and longitude. Each grid is indexed by the row ID and column ID. In Figure 2, we give the grid partition as an example where the ID pair of the Statue of Liberty is $(7, 9)$. The concatenation of the corresponding row embedding and column embedding makes up the representation of the grid.

In response to the issue of sparsity, we introduce a novel weighted BCE loss employing importance sampling. This strategy assigns greater weight to informative negative samples. They play a more significant role in influencing the gradient by assigning more weight to informative samples. As a result, the gradient's magnitude increases, thereby expediting the training process. Notably, this adaptive loss function can be seamlessly integrated with various negative sampling methods and optimized using diverse solvers. To further capitalize on geographic information, we introduce proximity-aware negative samplers. This approach involves the probabilistic selection of negative locations, giving a higher probability to more informative negative locations.

Overall, our main contributions are the following.

- We introduce the Proximity-Aware based region representation for Sequential Recommendation (PASR) built upon the Self-Attention Network, enhancing location recommendation. PASR effectively captures long-term sequential dependencies and optimally employs geographical data.
- We put up a novel self-attentive based geography encoder and a grid mapper. The encoder and mapper accurately represent the GPS coordinates of locations, enabling the capture of spatial proximity between nearby places. This approach models spatial clustering and distance-based location transitions better.
- We utilize the novel loss function that employs importance sampling to optimize PASR. This approach assigns higher weights to informative negative samples, which in turn accelerates training. We also utilize proximity-aware negative sampling method to enhance the quality of negative training instances.

The rest of the paper is organized as follows. We give some related work about sequential location recommendations, the methods for modeling geographical information, and

self-attentive based solvers for sequential recommenders. Next, some preliminaries about the attentive mechanism are followed in Section 3. Then, the details of the proposed model PASR are elaborated in Section 4, which includes the embedding layer, encoder-decoder module, and the negative sampler. In Section 5, we present abundant empirical studies to validate the superiorities of the proposed PASR. Finally, we conclude the paper in Section 6.

2. **Related Work.** In this section, we first concentrate on the advances of sequential location recommendation and then we illustrate the domain of sequential recommendation, especially the self-attention based solutions.

2.1. **Recommender for sequential location.** Recommender for sequential location has been approached through various modeling techniques, which includes interaction-based matrix factorization methods [6, 7], metric embedding techniques [8], word embedding methodologies [18], and Recurrent Neural Networks (RNNs) equipped with attention mechanisms [2, 9, 10, 11, 19]. The inclusion of geographical information has been achieved through the incorporation of travel distance [9, 10], the implementation of location transitions specific to distance [11], the utilization of geography-conscious uniform sampling [6], or the incorporation of geography-aware gating mechanisms within RNNs [19]. Temporal information has also been integrated through techniques such as embedding time intervals [10], time-of-the-week embedding [10, 20], and the regulation of information flow using interval-aware gating [19]. These models have been optimized using various methods, including the Bayesian Personalized Ranking (BPR) loss function [6, 7, 8, 9, 11, 21], cross-entropy loss [10, 19, 20], and hierarchical softmax [18]. The key distinctions between our proposed algorithm and these existing approaches encompass the geographic modeling techniques, the novel weighted loss utilization, and the incorporation of a self-attentive based network to capture long-term relationships.

2.2. **Methods for modeling geographical information.** The LBSN datasets have revealed the existence of spatial clustering patterns, which can be understood through Tobler's First Law of Geography [14, 22]. These patterns are further characterized by the distribution of distances between visited locations, displaying adherence to a power-law distribution. To circumvent assumptions related to power-law distributions, kernel density estimation has been utilized to estimate distance distributions between pairs of locations [23]. Nevertheless, modeling distance distributions may not fully account for the multi-center nature of one's visited places [24], prompting the development of geo-clustering methods aimed at identifying and grouping such locations [24, 25]. Estimating the number of clusters can be a challenging task, and as a solution, 2-D kernel density estimation has emerged to capture the spatial clustering phenomenon [13, 26, 27, 28]. These geographical modeling techniques have been incorporated into models in an ad-hoc manner.

The GPS coordinates present much more exact geographical information compared to modeling distribution of relative distance. Meanwhile, the sparsity of visited positions due to the power-law distribution brings challenges to model the geographical proximity. To overcome these defects, we introduce a self-attention-based geography encoder and a grid mapping approach. The encoder takes original GPS coordinates as input to fully utilize the geographical information. The proximity location can share a similar representation by the grid mapping mechanism which can relieve the sparsity of visited positions. The encoder and mapper can be compatible with the self-attentive network, offering a unique approach to capture geographical factors.

2.3. **Self-attentive based sequential recommender.** In this discussion, we will specifically focus on self-attentive based sequential recommendation. For a broader array of sequential recommendation algorithms, you may find more information in the provided survey references [29, 30]. The self-attention network, known for its complete parallelism and the capability to capture long-range dependencies, has been extensively employed in sequence modeling. It has delivered SOTA results over various domains, including Natural Language Processing (NLP) [31, 32, 33], combinatorial optimization [34, 35, 36], and social recommendation [37]. In recent years, this architecture has also been harnessed for sequential recommendation tasks, optimizing performance using the binary BCE loss based on inner product preferences [38] or the triplet margin loss based on l_2 distance preferences [39]. Empirical results show its significant performance improvement over traditional RNN-based methods. Nevertheless, the original self-attention network, initially designed for symbolic sequences, lacks inherent consideration for varying time intervals between consecutive interactions when modeling sequential dependencies [40, 41, 42, 43]. To address this limitation, the self-attention based network with time interval awareness was introduced [44], refining attention weights by incorporating time intervals. Moreover, to enhance recommendation performance while avoiding information leakage, an approach reminiscent of the Cloze task used in BERT training [45] was adopted [46], replacing the causality mask.

PASR sets itself apart from these methodologies through its innovative geography encoder and novel loss function. Combining these approaches could potentially lead to even more enhancements in recommendation accuracy.

3. **Preliminary: Attentive Mechanism.** To characterize the long-term relationship within sequences and better extract features from the sequence, we utilize the encoder based on self-attentive mechanism [38] as our sequence encoder to transform the input embedding matrix $\mathbf{E} \in \mathbb{R}^{m \times d}$ which maps a sequence with length m into d dimensional space. Specifically, the self-attention encoder is formed by stacking several self-attention modules, wherein each module comprises a Self-Attention (SA) layer and a Feed-Forward Network (FFN) layer.

The SA layer computes new representations for each element by considering relationships with all other elements in the sequence. In the SA layer, the input is taken from either the embedding matrix $\mathbf{E}$ or the hidden output from the previous self-attentive module (referred to as $\mathbf{E}$). This input is then transformed using three separate linear projection matrices: $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$. These transformations are then fed into an attentive mechanism to derive the final output:

$$\mathbf{S} = SA(\mathbf{E}) = Attention(\mathbf{E}\mathbf{W}_Q, \mathbf{E}\mathbf{W}_K, \mathbf{E}\mathbf{W}_V)$$

where the attention is in fact the scaled dot-product attention, i.e.,

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}$$

In this attention, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query, key, and value, respectively. Intuitively, it calculates a weighted sum over all values, where the weights are determined by the dot product of the query and the key. The scaling factor $\sqrt{d}$ is used to prevent the resulting dot product from becoming too large, which can lead to extreme weights, especially when the vector dimension d is large.

It is important to highlight that the predictive output for the $(n+1)$-th location relies solely on the preceding n behaviors, not future ones. To ensure adherence to this causality constraint, we enforce it through the use of a square mask. This mask is constructed with

$-\infty$ in its upper triangle and 0 in others. This approach effectively prevents information leakage from future behaviors into the prediction process. Although self-attention has the ability to adaptively gather information from a user's historical behaviors, it is inherently a linear model.

While self-attention can adaptively gather information from a user's historical behavior sequence, it is fundamentally a linear module. To introduce non-linearity to the module and contain interactions between different dimensions, we have incorporated the following feed-forward network layer into the self-attentive module:

$$\mathbf{F} = FFN(\mathbf{S}) = \max(0, \mathbf{S}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d_h}$, $\mathbf{W}_2 \in \mathbb{R}^{d_h \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{d_h}$, $\mathbf{b}_2 \in \mathbb{R}^d$, and they satisfy the condition $d_h > d$.

To capture more intricate features within the sequence, we stack N self-attention modules. However, as the neural network becomes deeper, issues such as vanishing gradients and slow training speed can arise. To ensure a stable training process and expedite training speed, we utilize residual connections and layer normalization [31] on every layer of the network. This leads to the formation of the final sequence encoder:

$$\mathbf{S}^l = \mathbf{F}^{l-1} + LayerNorm\big(SA\big(\mathbf{F}^{l-1}\big)\big)\mathbf{F}^l = \mathbf{S}^l + LayerNorm\big(FFN\big(\mathbf{S}^l\big)\big)$$

where $\mathbf{F}^0 = \mathbf{E}$ stands for the embedding matrix that acts as the input to the encoder, $\mathbf{S}^l$ refers to the output hidden of the l-th layer in self-attentive module, $\mathbf{F}^l$ represents the output hidden from the l-th layer of the FFN part, and the index $l \in \{1, 2, \ldots, N\}$.

In general, the input of the self-attentive based module is a sequence characterized by a matrix $\mathbf{E} \in \mathbb{R}^{m \times d}$ and the output is also a matrix with the same shape as $\mathbb{E}$. The main effect of self-attentive mechanism is to conduct the intra-interaction with the sequential information. In the following sections, the self-attentive mechanism based part is used in the Geography Encoder and self-attentive encoder.

4. PASR Method.

Let $L = \{l_1, l_2, \ldots, l_Q\}$ represent a set with Q locations, $U = \{u_1, u_2, \ldots, u_M\}$ represent a set of M users, and $S = \{s^{u_1}, s^{u_2}, \ldots, s^{u_M}\}$ represent a set of historical mobility sequences for all users. A sequential location recommender operates on a user's mobility trajectory, which is denoted as $S^u = r_1^u \to r_2^u \to \cdots \to r_n^u$. In this representation, $r_i^u = (u, l_i, \alpha_i, \beta_i)$ signifies a user behavior, where u indicates the user, l_i denotes the visited location, and α_i and β_i represent the latitude as well as the longitude of that location. Given this input trajectory, the objective of a sequential location recommender is to predict the subsequent location l_{i+1} along with its GPS position p_{i+1}. For training, the model takes into the user's trajectory excluding the final behavior $r_1^u \to r_2^u \to \cdots \to r_{n-1}^u$ as the input. The output hidden encompasses the trajectory excluding the initial behavior $r_2^u \to r_3^u \to \cdots \to r_n^u$. The representation module of the Point-of-Interest (POI) is depicted in Figure 3 and the general framework of the PASR is described in Figure 4. Each component of this architecture will be elaborated upon in the subsequent sections.

4.1. Embedding representation module for POI.

For user u, we can extract her/his interests from the historical mobility sequence $S^u = r_1^u \to r_2^u \to \cdots \to r_n^u$. For ease of parallelism, we preprocess the input sequence $r_1^u \to r_2^u \to \cdots \to r_n^u$ into a sequence with length m. If the original length of the input exceeds the given value m, we divide it into multiple sub-sequences, each with a length of m. In cases where the input sequence is shorter than m, we apply a "padding" operation at its right end, extending it until it reaches a length of m. To better represent the mobility sequence, three paradigms are utilized to embed the sequence, i.e., location ID embedding (w.r.t. Location Embedding

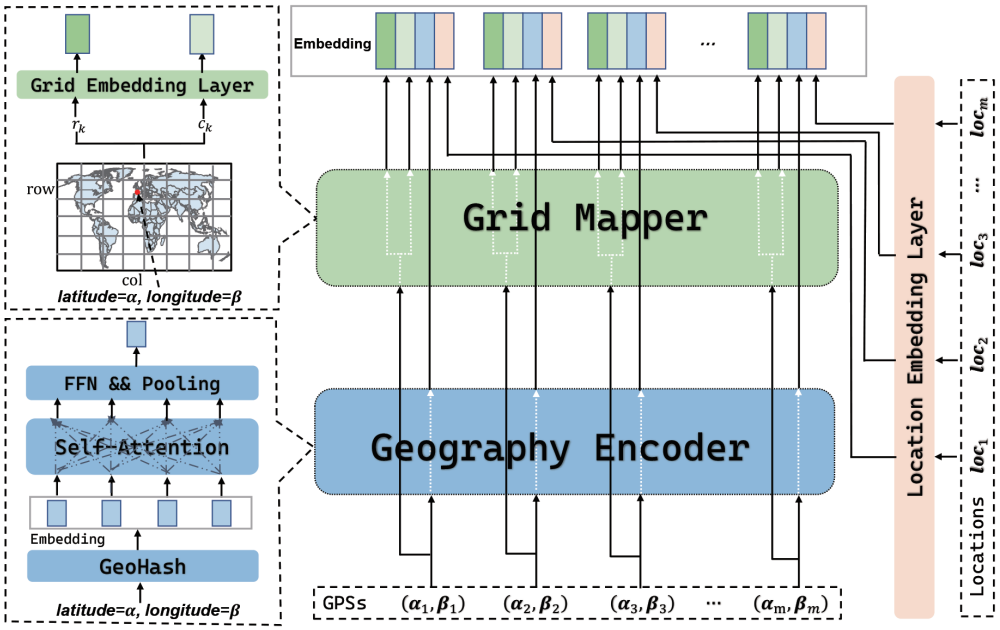

FIGURE 3. The embedding representation module of point-of-interest

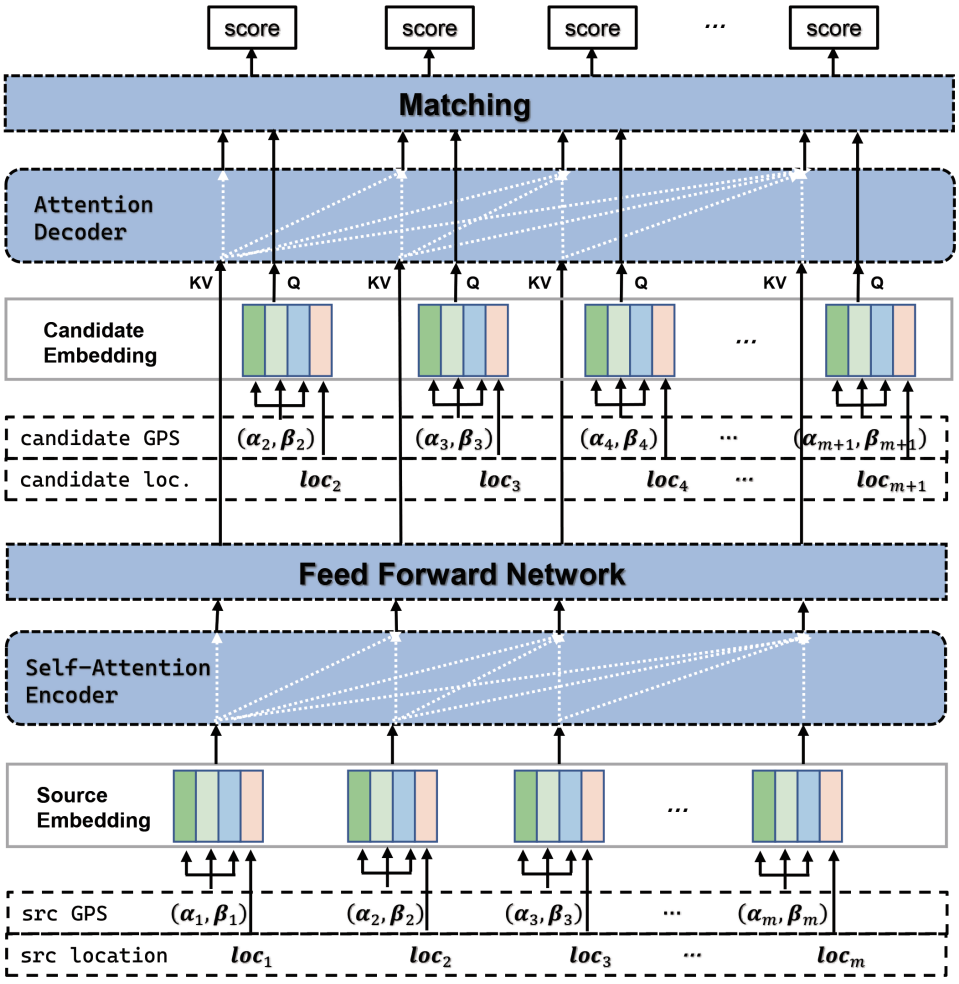

FIGURE 4. The overall framework of PASR

Layer), GeoHash embedding (w.r.t. Geography Encoder) and the proximity grid embedding (w.r.t. Grid Mapper) as depicted in Figure 3. In the embedding representation module, we directly take the location IDs and corresponding GPS coordinates as the input, which can fully utilize the geography information. The location embedding layer casts the one-hot id into the dense meaningful embedding, the geography encoder makes use of self-attention mechanism to model the relevance among the relative positions, and the grid mapper converts the gird where the POI is located into the geographic embedding to capture the proximity. Especially, we are the first one to combine two sub-embedding from two different embedding spaces to a grid embedding space, which can not only model the proximity but also save memory overhead.

4.1.1. *Location Embedding Layer.* Each location $l \in \{l_1, l_2, \ldots, l_Q\}$ corresponds to a one-hot embedding with dimension d, which comprises the location embedding matrix $\mathbf{M}_{loc} \in \mathbb{R}^{Q \times d}$. So the historical mobility sequence with length m can be represented by a matrix $\mathbf{E}_{loc} \in \mathbb{R}^{m \times d}$.

4.1.2. *Geography Encoder Layer.* The geographical information of a location is typically represented by its longitude and latitude coordinates. While it is theoretically possible to directly feed these floating-point values into the learning system, it is infeasible due to two reasons. First, as human activities involve a tiny fraction of the entire Earth's surface, using raw latitude and longitude values directly leads to severe sparsity problems. Second, latitude and longitude are closely intertwined because both are needed together to precisely determine a location. The complex interaction between these two attributes could pose a challenge for the learning system to fully comprehend and utilize. To tackle these challenges, we utilize the geography encoder as a solution.

For any input geography coordinate, the geography encoder first utilizes the GeoHash to dispose of it. The GeoHash system is a public-domain geocode that transforms a geographic coordinate into an alphanumeric string. This refers to a hierarchical spatial data structure that partitions space into grid-shaped buckets. Specifically, the entire map is first divided into 32 grids, and each sub-grid is assigned a number or letter as its identifier using the Base32 encoding scheme. Then, this process is applied recursively to the sub-grids until a designated precision level L is attained. At this stage, the entire map is divided into 32^L small grids, and each small grid is assigned a unique encoding string of length N composed of Base32 characters. The geographical coordinates of all locations within a grid can be denoted by the corresponding encoding string of that grid. For example, the GPS coordinates $(40.68925, -74.0445)$ of the Statue of Liberty are converted into a GeoHash string '$dr5r7p62n1$' through the GeoHash system (see Figure 1), with a string length specified as 12. The encoding process leverages both latitude and longitude, effectively managing their inherent interaction. Furthermore, regions in close proximity will have identical GeoHash strings, partially addressing the sparsity concern.

It is straightforward and intuitive to embed the encoding strings with an embedding matrix, but the number of encoding strings is exponential which may suffer from serious issues of sparsity, and the proximity relationships between the nearby locations cannot be captured effectively. To this end, we view an encoding string as a string of single characters, where each character belongs to the set of Base32 characters. It is important to note that a character-level module cannot thoroughly capture the proximity relationship among surrounding positions [47]. To overcome this limitation, we process every string into a sequence with n-grams. Thus, the vocabulary size expands from 32 to 32^n. For instance, let us consider the initial 6 characters – '$dr5r7p$' of the aforementioned string. It is transformed into a sequence of bigrams: $dr \to r5 \to 5r \to r7 \to 7p$. Once we obtain the n-gram sequence, we embed this sequence and subsequently apply a stacked self-attentive

based network to characterizing sequential dependencies. Following this, we aggregate the representations of the n-gram sequence using an average pooling technique, seeing the left-down subfigure of Figure 3. In this way, each geography coordinate of location in the historical mobility sequence can be embedded into a vector with d dimension by GeoHash encoding and Goegraphy Encoder.

4.1.3. *Grid Mapper.* After applying the geography encoder to the GPS coordinate, we have obtained a good representation of geography information of the location. However, note that the using of GeoHash still has some limitations. First, the GeoHash system could be faced with edge cases, where two close locations have completely different GeoHash strings. For example, the distance between GPS coordinates $(-0.005, 90)$ and $(0.011, 90)$ is only about 1.78 km, but their corresponding encoding string is '*qpbpbp*04*b*5*bj*' and '*w*00004000481', respectively. Second, although the use of n-gram in transforming the encoding string could improve the quality of the representation of geography information, the larger vocabulary size may exacerbate the sparsity problem. To solve the challenges mentioned above, we propose using the grid mapper to reprocess the latitude and longitude, which will make better use of the geography information and can be viewed as a supplementary of the geography encoder.

The grid mapper only considers the valid region that contains locations and then divides the map into grids based on latitude and longitude. Concretely, we first traverse the GPS coordinates of all locations and record the highest and lowest values of latitude and longitude respectively during pre-processing. In the subsequent steps, we only focus on the region bounded by these recorded maximum and minimum values of latitude and longitude, which helps alleviate sparsity issues. We treat this region as a flat surface and divide it into grids along both latitude and longitude. Latitude corresponds to rows, while longitude corresponds to columns. With this setup, the region is divided into numerous grids, and each grid is indexed by the row ID and column ID (see Figure 2). Any given location falls within one of these grids, and its GPS coordinates are represented by the ID pairs of the corresponding grid. Following this, we directly embed the row ID and column ID into dense vectors with dimension d, seeing the right-top subfigure of Figure 3.

4.1.4. *Representation of the historical mobility sequence.* As the aforementioned description, we apply a location embedding matrix $\mathbf{M}_{loc} \in \mathbb{R}^{Q\times d}$, where d denotes the feature vector dimension, to converting one-hot encodings of interest points into dense embedding vectors. Thus, after applying the $\mathbf{M}_{loc}$ matrix transformation to the input sequence of length m, we obtain an embedding matrix $\mathbf{E}_{loc} \in \mathbb{R}^{m\times d}$ to serve as subsequent input. For GPS sequences with length m composed of latitude and longitude, we utilize a geography encoder and a grid mapper to transform them into a geographic embedding matrix $\mathbf{E}_{geo} \in \mathbb{R}^{m\times d}$, a row embedding matrix $\mathbf{E}_{row} \in \mathbb{R}^{m\times d}$, and a column embedding matrix $\mathbf{E}_{col} \in \mathbb{R}^{m\times d}$. We now have obtained the location embedding matrix $\mathbf{E}_{loc}$, the geographic embedding matrix $\mathbf{E}_{geo}$, the row embedding matrix $\mathbf{E}_{row}$, and the column embedding matrix $\mathbf{E}_{col}$. These four matrices are concatenated to form the embedding matrix $\mathbf{E}_{input} \in \mathbb{R}^{m\times 4d}$, which serves as input to the following network. As we use a sequence encoder based on self-attention to transform the input matrix, it differs from recurrent or convolutional modules and cannot capture positional information in the sequence. Similar to the approach in [38], we add a learnable positional embedding matrix $\mathbf{P} \in \mathbb{R}^{m\times 4d}$ to the input embedding matrix $\mathbf{E}_{input}$, that is, $\mathbf{E}_{input} = \mathbf{E}_{input} + \mathbf{P}$.

4.2. **Attentive-based encoder-decoder.** The overall framework of PASR is illustrated in Figure 4. For a user, the corresponding historical mobility sequence is fed into the embedding representation module (see Figure 3) to obtain the source embedding. Then the

source embedding is fed into the self-attentive based encoder followed by an FFN to get the hidden representation (denoted as $\mathbf{F}^N \in \mathbb{R}^{m\times 4d}$) of the historical mobility sequence. As we aim to predict the next location, the historical mobility sequence consists of the next locations (i.e., the sequence consists of locations from second to $m+1$-th location) is regarded as the candidate target and the corresponding candidate embedding (denoted as $\mathbf{T} \in \mathbb{R}^{m\times 4d}$) as derived according to the embedding representation module (a.k.a. Figure 3). Our proposed PASR model takes the raw POI identifiers and GPS coordinates as inputs, utilizing the prowess of the advanced self-attention mechanism intertwined with a versatile encoder-decoder framework, which captures the semantic information and geographic relationships among different POIs better and achieves superior recommendation performance.

Many of the current recommendation systems based on self-attentive mechanism directly pass the encoder's outputs to the matching module. However, recent research findings [48, 49] suggest that this approach may not be optimal. To enhance the representative quality of the input sequence in relation to target positions, the PASR model introduces a decoder equipped with target-aware attention. The mechanism of this decoder is described as

$$\mathbf{A} = \mathit{decoder}\left(\mathbf{F}^N \middle| \mathbf{T}\right) = \mathit{Attention}\left(\mathbf{T}, \mathbf{F}^N\mathbf{W}, \mathbf{F}^N\right)$$

where $\mathbf{T} \in \mathbb{R}^{m\times 4d}$ is the representation matrix of the output sequence and $\mathbf{W} \in \mathbb{R}^{4d\times 4d}$ serves the purpose of projecting queries and keys into a common latent space. Importantly, it should be noted that the causality constraint remains a necessity, which is accomplished through the utilization of the mask as previously mentioned.

Prediction scores. With the representation of the input behaviors (i.e., $\mathbf{A}_i$) at the step i, the preference scores of all the candidate locations can be computed using any matching function f. This matching function could be a deep neural network [49], the dot product [38] when the historical and candidate representations have identical dimensions, or a bilinear function when their dimensions differ. Concretely, preference scores are formulated as follows:

$$y_{i,j} = f(\mathbf{A}_i, \mathbf{T}_j)$$

where $\mathbf{T}_j$ represents the feature vector for candidate location j, derived by concatenating the embedding vectors of that location and the feature vectors of its geographical information. Just like in [38], the embedding matrices, the geography encoder, and the grip mapper are common to both the input sequences and the output sequences.

4.3. **Importance sampling based loss.** Given the sequence S^u, the candidate location j's preference score at step i is denoted as $y_{i,j}$. It is clear that optimizing the widely adopted cross-entropy loss function [10, 20, 49] is inefficient in this scenario when there is a remarkable increase in the size of candidate locations. In scenarios where self-attention based sequential recommenders are concerned, the Binary Cross-Entropy (BCE) loss is commonly adopted [38, 44]. This loss is written as follows:

$$\mathcal{L} = -\sum_{S^u\in S}\sum_{i=1}^{n}\left(\log\sigma\left(y_{i,t_i}\right) + \sum_{l\notin L^u}\log\left(1-\sigma\left(y_{i,l}\right)\right)\right)$$

where S represents the set consisting of users' mobility trajectories, L^u means the set of positions visited by user u, and t_i signifies the target position at time step i. In this context, we have already excluded the padding item from the loss calculation. For effective loss optimization, one negative location is randomly selected from the unvisited locations for user u at each time step.

As only one negative location is randomly selected from the user's unvisited locations, the BCE loss may not efficiently leverage the abundance of unvisited locations. Specifically, once the loss has been optimized for a few epochs, positive locations become readily distinguishable from randomly chosen negative samples. Consequently, the gradient of the loss decreases dramatically, leading to sluggish training progress. Essentially, the absence of informative negative samples obstructs the optimization of the BCE loss. Intuitively, unvisited locations with high preference scores should exert a more substantial influence on the gradient. These locations inherently possess more valuable information and should be sampled with higher probabilities. Nevertheless, directly selecting the top-k unvisited locations with the greatest scores as negatives is not feasible due to the risk of introducing false negatives. And directly sampling negative locations in proportion to their preference scores is also infeasible due to the efficiency challenges. To address these challenges, we adopt the similar approach proposed in [50] to assign weights to unvisited locations based on negative probabilities. By employing this method, we ensure that even when using a uniform sampler, locations with more information can receive enhanced attention. Next, the BCE loss is

$$\mathcal{L} = -\sum_{S^u \in S} \sum_{i=1}^{n} \left(\log \sigma(y_{i,t_i}) + \sum_{l \notin L^u} P(l|i) \log(1 - \sigma(y_{i,l})) \right)$$

where $P(l|i)$ represents the probability of the position l being negative instance when we know the user u's visited positions are $r_1^u \to r_2^u \to \cdots \to r_i^u$. We suggest modeling the probability being negative as follows:

$$P(l|i) = \frac{\exp(y_{i,l}/T)}{\sum_{l' \notin L^u} \exp(y_{i,l'}/T)}$$

in which, T represents a temperature that controls the deviation of the proposal distribution from a uniform one. As T tends towards infinity, the distribution approaches uniformity.

Despite this, the efficiency of the restructured loss function is still hampered by the computational burden of normalizing probabilities. To enhance efficiency, while accounting for $\sum_{l \notin L^u} P(l|i) \log(1 - \sigma(y_{i,l}))$, we adopt the notion of expectation computation in relation to $P(l|i)$. We introduce an approach that approximates this expectation through importance sampling. Let us assume a proposal distribution denoted as $Q(l|i)$ from which simple sampling can be carried out. Furthermore, we represent $\tilde{Q}(l|i)$ as the unnormalized probability of $Q(l|i)$. With inspiration drawn from [51], the loss is then approximated in the subsequent manner:

$$\mathcal{L} = -\sum_{S^u \in S} \sum_{i=1}^{n} \left(\log \sigma\left(y_{i,t_i}\right) + \sum_{l \notin L^u} w_l \log\left(1 - \sigma\left(y_{i,l}\right)\right) \right)$$

where $w_l = \frac{\exp\left(y_{i,l}/T - \ln \tilde{Q}(l|i)\right)}{\sum_{l'=1}^{l} \exp\left(r_{i,l'}/T - \ln \tilde{Q}(l'|i)\right)}$ is the weight of the corresponding sample. Consequently, within the set consisting of l locations, those with higher preference scores are allocated greater weights. In scenarios where the proposal distribution Q departs from the distribution P, the weight serves to mitigate the deviation between P and Q to a certain degree.

It is important to highlight that we exclusively make use of the unnormalized probability, a choice that proves advantageous when working with probability distributions that pertain to a subset of the overall location set L. In cases where the proposal distribution $Q(l|i)$ assumes a uniform distribution over $L \backslash L^u$, we observe that $\ln \tilde{Q}(l|i) \propto -\ln |L|$,

which results in the weight $w_l = \frac{\exp\left(y_{i,l}\right)/T}{\sum_{l'=1}^{K}\exp\left(y_{i,l'}/T\right)}$. Approximating $Q(l|i)$ with the uniform distribution over L is a valid approach, primarily due to the minimal probability associated with sampling positions from L^u as negative samples and the comparable level of recommendation accuracy. When devising alternative proposal samplers, we adopt a similar strategy, considering the distribution over L instead of $L\backslash L^u$ to simplify the process. This method is widely employed in the field of NLP [52].

4.4. **Proximity-aware negative sampler.** Geographical information is crucial for a sequential location recommender as it helps differentiate between negative and potentially positive unvisited locations. For instance, when a user visits the location t, the unvisited locations proximate to t may be more inclined to be negative. Nevertheless, directly sampling locations grounded in GPS distance might be computationally infeasible. In response to this challenge, the proximity-aware negative samplers adopt a pragmatic approach. Given a target location, we first retrieve the nearest K locations. Subsequently, negative samples are randomly drawn from these K candidates. The negative sampling procedure can be based on either a uniform distribution or a distribution based on popularity. In the context of the popularity-based distribution, the empirical results underscore the efficacy of utilizing $\tilde{Q}(l|i) \propto \ln(c_l + 1)$, where c_l represents the occurrence frequency in the mobility history.

Our proposed distribution formulation can effectively guide the negative sampling process.

5. **Experiment.** In this section, several SOTA baselines are compared with the proposed **PASR** methods on three real-world datasets.

5.1. **Datasets.** We evaluated our method by utilizing three public LBSN datasets: Weeplaces, Gowalla, and Brightkite, which are collected from real-world APPs. These APPs record the user's trajectories like shopping places, restaurants, and entertainment venues. These datasets are also widely used in most POI recommendation literature, like [53, 54, 55]. These commonly used geographical location datasets contain significant amounts of user location and behavior data, which are valuable for location-based recommendations and urban planning. Below is a brief description of these three datasets and their sources.

The Gowalla dataset is provided by the Gowalla social networking platform, which operated from 2007 to 2012. The Gowalla dataset includes information like the geographic locations of user's check-ins, timestamps, and the points of interest they checked into. It is a large-scale geographical location dataset, comprising millions of check-in records and hundreds of thousands of points of interest.

Launched in 2007 and subsequently disbanded in 2012, Brightkite is a social media network based on position records. The platform allowed users to check in at locations after visiting them through text messages or mobile applications. It is a moderately large geographical location dataset, containing millions of check-in records and tens of thousands of points of interest.

Weeplaces visualizes users' check-in behavior on maps and has been integrated into several LBSN APIs. Users can log in to Weeplaces using their accounts from LBSNs and connect with friends who have also used the application on the same LBSN. It is a medium-sized dataset comprising millions of check-in records and tens of thousands of points of interest.

Similar to the approach in [3], we conducted data filtering by removing users with less than 20 check-ins and locations with less than 10 visits. Table 1 summarizes the overall information for these datasets after this filtering process. For each user, we constructed a

check-in sequence based on their check-in records in chronological order. For every user's check-in sequence, we selected the user's last check-in at an unvisited point of interest before that moment for testing and used the check-in records before that for training. This ensures that the evaluation is focused on predicting locations that the user has not visited in their historical records, making it more consistent with real-world scenarios. The length of the sequence was set to 50. If a sequence exceeded 50, we divided it into non-overlapping sub-sequences of length 50 and used the most recent 50 check-in records for testing.

Table 1. Statistics of datasets

	Gowalla	Brightkite	Weeplaces
#locations	131,327	48,177	127,859
#users	31,708	5,186	13,801
#check-ins	2,963,324	1,661,161	5,328,782

5.2. **Baseline methods.** The following baseline models are compared with PASR.

BPR [17]: A recommendation algorithm based on Bayesian personalized ranking, primarily used for handling implicit feedback data. The aim of the BPR is to learn a ranking function for each user that places items they prefer above items they dislike. BPR optimizes its loss function by maximizing posterior probabilities, resulting in low-dimensional latent vectors for users and points of interest (POIs). These latent vectors are used to calculate user preference scores for POIs and make recommendations.

FPMC [6]: A conventional recommendation system algorithm that combines ideas from matrix factorization and Markov chains. In FPMC, the recommendation process is modeled as a sequence of events or transitions between items, and it leverages users' historical interactions to make recommendations.

GRU4Rec [56]: The variant of RNNs which is commonly used to handle sequential data. In POI recommendation, GRU can be applied to model users' historical visit sequences to capture user interest evolution and temporal dependencies. By inputting a user's historical visit sequence into GRU, the model can learn similarities between users and associations between POIs, enabling it to make recommendations.

SASRec [38]: A sequence recommendation model based on self-attention mechanisms, particularly suitable for handling user sequential behavior data. It encodes a user's historical check-in sequence into vector representations and employs self-attentive mechanisms to characterize the relationships between different items in the sequential data. This allows the model to consider both previously visited POIs and the current context information during recommendation, resulting in personalized recommendations.

GeoSAN [50]: A sequence recommendation also uses self-attention mechanisms and incorporates geography information in the model. It encodes a GPS coordinate to a quadkey string and then uses a self-attention based encoder to convert the quadkey string to an embedding with geographical information. The representation of one POI is the concatenation of its id embedding and the geographical embedding. The major difference between GeoSAN and our proposed model is the use of the GeoHash algorithm and the introduction of the grid mapper.

5.3. **Metric.** The capability of recommendations is estimated by the ranking of interest points that users actually visited within the recommended list provided by the recommendation system. We assess our POI recommendation model using two widely used evaluation metrics in recommendation systems: Hit Ratio (HR@k) and Normalized Discounted Cumulative Gain (NDCG@k).

HR@k (Hit Ratio) refers to the proportion of users, whose true interest points appear in the first k items of the recommended candidates generated for a user. It emphasizes the precision of the recommended results, i.e., whether the items of interest to the user appear in the recommended list. It is written as follows:

$$\text{HR@k} = \sum_{u \in U} \frac{I(R_u \cup T_u \neq \emptyset)}{|U|}$$

where U represents all the users, R_u denotes the results recommended for user u, T_u denotes the ground truth set containing relevant items w.r.t. user u, and $I(\cdot)$ denotes the indicator function.

NDCG@k (Normalized Discounted Cumulative Gain) refers to the cumulative gain of relevant items in the top k positions of a recommended list, divided by the maximum possible cumulative gain under ideal conditions. It emphasizes the utility of the model's recommendations, i.e., whether relevant items are ranked reasonably in the recommended list. NDCG@k can handle differences in relevance between different items, giving higher weight to more relevant items. Its formula is as follows:

$$\text{NDCG@k} = \frac{\text{IDCG@k}}{\text{DCG@k}}$$

where DCG@k represents the Discounted Cumulative Gain for the top k positions in the recommended list, taking into account the positional influence. It is described as

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$$

where rel_i is the relevance score of the recommended item at position i. In experiments, we set the relevance score for relevant locations as $\text{rel}_i = 1$ and for irrelevant locations as $\text{rel}_i = 0$. IDCG@k represents the Ideal DCG@k, which is the maximum possible DCG@k for a recommended list sorted by relevance scores in descending order.

A set of 100 locations is randomly chosen to serve as negative candidates for ranking alongside the target location. Subsequently, we can calculate the Hit Rate and NDCG metrics by the ranking results of the recommended 101 positions.

5.4. **Setting.** We conduct coarse grid search over the hyperparameters space to determine the setting of hyperparameters. We set the dimensions of interest point embedding vectors, geographic embedding vectors, grid row embedding vectors, and grid column vectors as $d = 50$, the latent space dimension d_h as 128, and the number of layers for the self-attention layers in the sequence encoder, geography encoder, and target-aware attention decoder as 2. The number of intervals in the grid mapper is set to 5000, i.e., we divide the valid region into $5000 * 5000$ grids. And we use a 3-gram token to represent the GeoHash string to utilize the geography information. For each location within the dataset, we initially retrieve the $K = 2000$ nearest locations for the proximity-aware negative sampling. The number of negative training samples is set to 5 on all three datasets. To train the proposed model, we use Adam optimizer, with a learning rate of 0.001 and weight decay of 0.0001. We perform 20 epochs of training on each dataset. For the benchmark models we are comparing against, in order to facilitate a fair comparison, we set their model parameters and training processes the same wherever applicable.

5.5. **Comparison with baselines.** Table 2 presents the HR and NDCG metrics for our model and benchmark models on three datasets. Overall, it can be observed that the performance in Point-of-Interest (POI) recommendation tasks progressively improves from BPR, FPMC, GRU, SASRec, and GeoSAN, to our proposed model.

TABLE 2. Comparison with baselines

Dataset	Metric	BPR	FPMC	GRU4Rec	SASRec	GeoSAN	PASR
Gowalla	HR@5	0.0872	0.1783	0.1830	0.2768	0.3311	**0.3591**
	NDCG@5	0.0523	0.1058	0.1082	0.1977	0.2378	**0.2522**
	HR@10	0.1618	0.3471	0.3536	0.3887	0.4610	**0.5096**
	NDCG@10	0.0762	0.1597	0.1627	0.2336	0.2795	**0.2969**
Brightkite	HR@5	0.0850	0.1873	0.1908	0.3105	0.4059	**0.4238**
	NDCG@5	0.0521	0.1119	0.1144	0.2027	0.2897	**0.2977**
	HR@10	0.1479	0.3537	0.3634	0.4285	0.5607	**0.5642**
	NDCG@10	0.0721	0.1651	0.1695	0.2374	0.3398	**0.3431**
Weeplaces	HR@5	0.0812	0.1778	0.1843	0.2527	0.3011	**0.3207**
	NDCG@5	0.0482	0.1055	0.1090	0.1834	0.2099	**0.2261**
	HR@10	0.1526	0.3458	0.3557	0.3513	0.4397	**0.4555**
	NDCG@10	0.0710	0.1592	0.1637	0.2150	0.2544	**0.2694**

BPR performs poorly in the POI recommendation task, likely due to its reliance on training solely based on the user's implicit feedback, failing to capture sequential effects in user check-in behaviors. On the other hand, FPMC models the transition relationships effectively using Markov chains, thus significantly improving recommendation performance compared to BPR. GRU captures sequential effects in user check-in behaviors through its recurrent neural network structure, resulting in some performance improvement. SASRec effectively captures both short-term and long-term relationships in sequences benefiting from self-attention mechanisms, leading to improved recommendation performance. GeoSAN, being a strong benchmark model, based on self-attention mechanisms and utilizing hierarchical gridding of GPS locations to effectively leverage geographic information, achieves superior recommendation performance. Comparing PASR to GeoSAN, we observe a 4.41% to 8.46% improvement in the HR@5 metric and a 2.76% to 7.72% improvement in the NDCG@5 metric. This demonstrates the effectiveness of PASR.

5.6. **Ablation study.** To assess the impact of different elements within our approach, we carried out the ablation study. The basic method (PASR) adopts the structure illustrated in Figure 4 and takes into account the following model variants for comparison.

- *US (Uniform Sampler)*: Instead of using the proximity-aware negative sampling method, a uniform sampler over all unvisited locations is employed during training.
- *BCE Loss*: The vanilla BCE loss, i.e., the weights of negative instances are one, is used and essentially reverting to a simpler loss function.
- *Remove GE*: The geography encoder was removed from the model architecture, and the representation is constructed by concatenating the location embedding, grid row embedding, and grid column embedding.
- *Remove GM (Grid Mapper)*: We remove the grid mapper and use the concatenation of location embedding and geographic embedding as the representation.
- *Remove GE (Geography Encoder) and GM (Grid Mapper)*: Both the geography encoder and grid mapper are simultaneously removed, leaving only the location embedding as the representation.
- *Remove AD (Attention-based Decoder)*: The attention-based decoder is removed from the model, and only the output hidden of the encoder is used for matching.
- *Add UE (User Embedding)*: Each user is transformed into a dense embedding, which is then added to the location embedding.

- *Add TE (Time Embedding)*: Timestamps of check-in records are shining upon one-hour intervals within one week (resulting in 168 time intervals), and finally these intervals are put into an embedding layer to obtain time embeddings.

The performance of the ablation study is summarized in Table 3. From the table, the following findings are summarized.

- I. *The proximity-aware negative sampler is helpful in certain circumstances.* Employing the proximity-aware negative sampler benefits the recommendation performance on the datasets Gowalla and Weeplaces, but leads to a slight decline in dataset Brightkite. The reason behind that is maybe the number of locations in the Brightkite is relatively small, and the using of the proposed sampler may lead to sample some false negative samples.
- II. *The proposed novel loss function demonstrates its efficacy.* Compared to the result where vanilla BCE loss is employed, our proposed loss can improve the performance by 1.69%, 0.71%, and 2.61% on the term of NDCG@5. This is because the proposed new loss can assign higher weight to the more informative negative samples to promote the training process.
- III. *The utilization of geography information plays a significant role in improving the recommendation performance.* We use a geography encoder and grid mapper to utilize the geography information, and removing any part of them will lead to the degradation of recommendation accuracy. It is notable that the using of a geography encoder and grid mapper can improve the performance dramatically, and the improvements on the three datasets are 8.61%, 12.51%, and 10.45% in terms of NDCG@5. It is interesting that only removing one of the geography encoder and grid mapper will only lead to a slight decline, which means that both the geography encoder and grid mapper can incorporate the geography information well.
- IV. *Adding user embedding or time embedding plays no role in recommendation accuracy.* Adding time embedding only improves the NDCG@10 slightly on the Gowalla, while adding user embedding will cause dramatic performance degradation on the three datasets. This may because the introduction of time embedding or user embedding will lead to a mismatch between the candidate embedding space and the check-in embedding space.
- V. *Using an attentive-based decoder will improve the performance.* The improvements are 1.21%, 1.65% and 4.53% on the three datasets in terms of HR@5. The decoder makes the output of encoder to interact with the historical check-ins which are related to the target locations, resulting in enhancements in recommendation performance.

5.7. **Performance w.r.t. loss function and negative sampling.** Different combinations of loss function and negative samplers are investigated. Concretely, we experimented on the following four combinations: ① weighted loss (our proposed loss) + kNN-uniform sampler (our proposed sampler); ② unweighted loss (the vanilla BCE loss) + uniform sampler (sampling from all unvisited locations uniformly); ③ unweighted loss + kNN-uniform sampler; ④ weighted loss + uniform sampler. For each combination of the loss and the negative sampling method, the number of negative instances is varied from 1 to 8. Experimental results on the Gowalla dataset are presented in Figure 5.

We can make the following observations. First, the model trained based on the kNN-uniform sampler outperforms the model trained with the uniform sampler. This demonstrates the effectiveness of the proposed negative sampler. By sampling based on proximity, the kNN-sampler can select more informative negative instances, thereby enhancing the training process. Second, the weighted loss contributes significantly to model performance. The impact of the weighted loss is particularly pronounced when using the

TABLE 3. Ablation study results

Dataset	Metric	US	BCE	−GE	−GM	−GE-GM	−TAAD	+UE	+TE	PASR
Gowalla	HR@5	0.3494	0.3590	0.3554	0.3537	0.3328	0.3548	0.3313	0.3581	**0.3591**
	NDCG@5	0.2485	0.2480	0.2478	0.2474	0.2322	0.2520	0.2323	0.2505	**0.2522**
	HR@10	0.4827	0.5087	0.5026	0.4988	0.4684	0.4923	0.4699	0.5026	**0.5096**
	NDCG@10	0.2915	0.2962	0.2953	0.2941	0.2758	0.2964	0.2770	**0.2971**	0.2969
Brightkite	HR@5	**0.4288**	0.4142	0.4128	0.4057	0.3712	0.4169	0.3803	0.4136	0.4238
	NDCG@5	**0.3049**	0.2956	0.2935	0.2916	0.2646	0.2952	0.2642	0.2929	0.2977
	HR@10	0.5613	0.5453	0.5553	0.5461	0.4952	0.5565	0.5162	0.5561	**0.5642**
	NDCG@10	**0.3477**	0.3410	0.3395	0.3366	0.3047	0.3402	0.3080	0.3387	0.3431
Weeplaces	HR@5	0.3126	0.3155	0.3048	0.2956	0.2891	0.3068	0.2940	0.3121	**0.3207**
	NDCG@5	0.2196	0.2202	0.2151	0.2073	0.2047	0.2179	0.2055	0.2192	**0.2261**
	HR@10	0.4456	0.4506	0.4362	0.4362	0.4198	0.4412	0.4251	0.4544	**0.4555**
	NDCG@10	0.2624	0.2668	0.2574	0.2526	0.2467	0.2612	0.2476	0.2648	**0.2694**

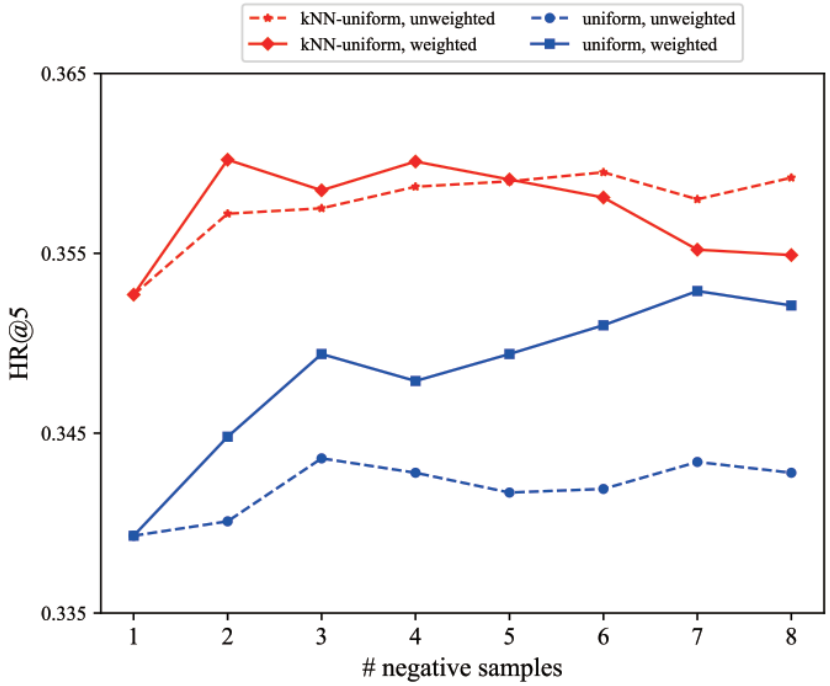

FIGURE 5. The performance when using different negative sampling methods, different number of negative instances, and different losses

uniform sampler. When employing the kNN-sampler, the weighted loss facilitates performance improvement with a small size of negative instances. However, as the size of negative instances exceeds a certain threshold, performance may experience a slight decline. Such degradation could result from assigning higher weights to irrelevant or noisy negative samples as the number increases. Hence, selecting an appropriate number of samples is crucial for training.

5.8. **Sensitivity w.r.t. embedding dimension.** The dimension of embedding is set to be the value from 10 to 60 with step size 10. Figure 6 exhibits the results. We observe a significant deterioration in performance when employing a small embedding dimension, whose expressive power is limited. Optimal recommendation accuracy is achieved with a moderate embedding size of 30, which can effectively capture the inherent meaning of

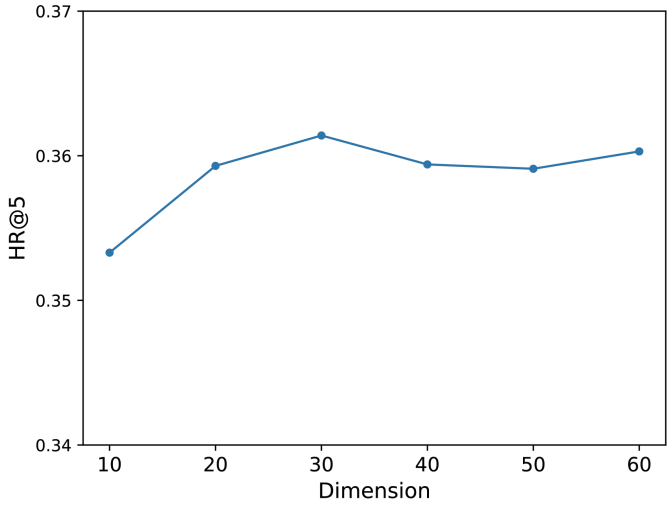

FIGURE 6. The performance w.r.t. embedding dimension

location and geographical data. Increasing the embedding dimension further, however, adversely impacts recommendation accuracy, as the number of locations and regions is limited.

5.9. **Sensitivity analysis w.r.t. different sizes of interval in grid mapper.** The number of intervals used in the grid mapper is varied from 3000 to 8000 with a step 1000. Figure 7 shows the results. The peak performance is obtained when the interval number is a medium value of 6000. Too low or too high a number of intervals can cause performance degradation. The former is due to lacking of differentiation between different regions, and the latter is because a large number of intervals means the grid mapper divides the map into a greater number of regions which could cause the sparsity problem. In general, the proposed model is insensitive to the interval number in the grid mapper.

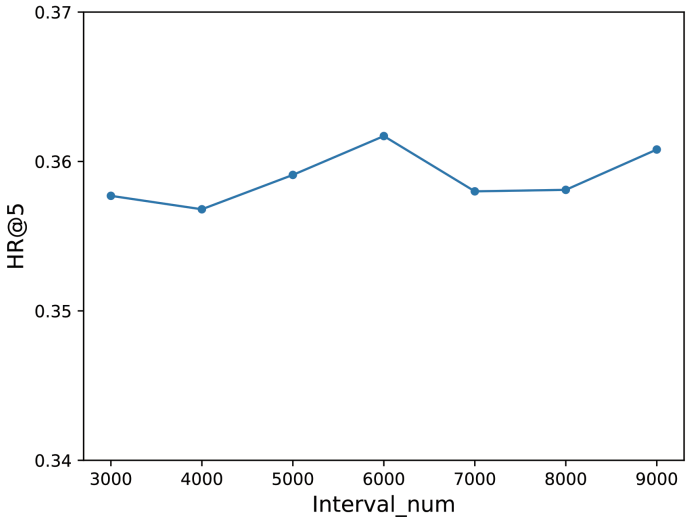

FIGURE 7. The performance w.r.t. different sizes of interval in grid mapper

5.10. **Sensitivity w.r.t. *n*-gram.** We vary the n-grams from $n = 1$ to $n = 6$. Figure 8 shows the results. The peak performance is obtained when $n = 2$. The size of gram token vocabulary is 32^n, which will increase very rapidly as n increases. The large n will make

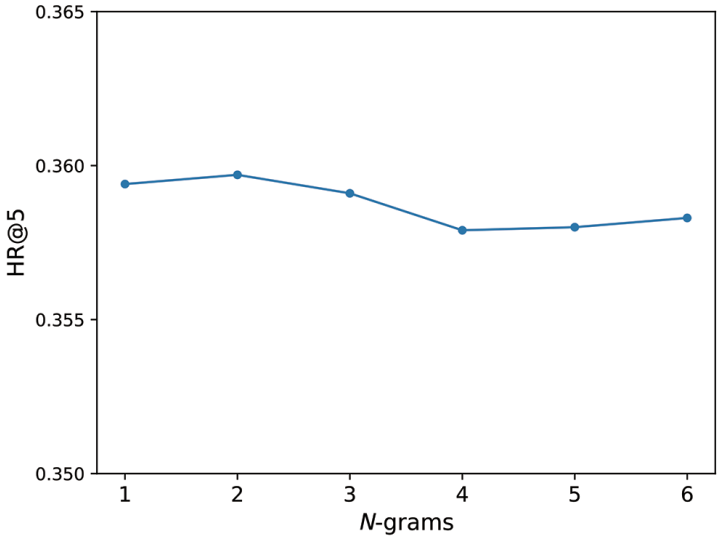

FIGURE 8. The impact of n-gram

the vocabulary size too large, and we can observe a slight decline of recommendation accuracy. In general, the proposed model is insensitive to n, and we recommend using a smaller n for both better performance and smaller memory cost.

6. **Conclusions.** In this paper, we put up a self-attentive based model PASR for sequential location recommendation. To address the unbalance between positive instances and negative instances and utilize the information from hard negative samples, we apply the loss function based on importance sampling. Additionally, we have designed a novel geography encoder that incorporates geographical information and proximity property among positions, allowing us to implicitly capture distance-aware location transitions and spatial clustering phenomena. Moreover, to extract more information from negative instances, we proposed to use proximity-aware negative samplers. The new model PASR is tested on three large-scale datasets. The empirical results exhibit that PASR outperforms the state-of-the-art sequential location recommendation method significantly. Through the ablation study and sensitivity analysis, we also demonstrate the effectiveness of the loss based on importance sampling, the geography encoder, the grid mapper, and the proximity-aware negative sampler at improving recommendation performance. The proposed PASR model is to rank the candidate POI with a modest candidate size. As the whole candidate space is large only the rank phase cannot adopt the exposure of candidate size improving. In the future, the retrieve and rank phases can be fused into the same framework and trained jointly for a better recommendation experience.

Acknowledgment. This work is supported by Science Research Foundation of Shenzhen Institute of Information Technology (11400). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] M. C. Gonzalez, C. A. Hidalgo and A.-L. Barabasi, Understanding individual human mobility patterns, *Nature*, vol.453, no.7196, pp.779-782, 2008.

[2] Y. Wu, D. Lian, S. Jin and E. Chen, Graph convolutional networks on user mobility heterogeneous graphs for social relationship inference, *Proc. of the 28th International Joint Conferences on Artificial Intelligence*, pp.3898-3904, 2019.

[3] D. Lian, Y. Zhu, X. Xie and E. Chen, Analyzing location predictability on location-based social networks, *Proc. of Advances in Knowledge Discovery and Data Mining: The 18th Pacific-Asia Conference*, pp.102-113, 2014.
[4] C. Song, Z. Qu, N. Blumm and A.-L. Barabási, Limits of predictability in human mobility, *Science*, vol.327, no.5968, pp.1018-1021, 2010.
[5] S. Rendle, C. Freudenthaler and L. Schmidt-Thieme, Factorizing personalized Markov chains for next-basket recommendation, *Proc. of the 19th International Conference on World Wide Web*, pp.811-820, 2010.
[6] C. Cheng, H. Yang, M. R. Lyu and I. King, Where you like to go next: Successive point-of-interest recommendation, *Proc. of the 23rd International Joint Conference on Artificial Intelligence*, pp.2605-2611, 2013.
[7] D. Lian, V. W. Zheng and X. Xie, Collaborative filtering meets next check-in location prediction, *Proc. of the 22nd International Conference on World Wide Web*, pp.231-232, 2013.
[8] S. Feng, X. Li, Y. Zeng, G. Cong and Y. M. Chee, Personalized ranking metric embedding for next new POI recommendation, *Proc. of the 24th International Conference on Artificial Intelligence*, pp.2069-2075, 2015.
[9] Q. Cui, Y. Tang, S. Wu and L. Wang, Distance2Pre: Personalized spatial preference for next point-of-interest prediction, *Proc. of Advances in Knowledge Discovery and Data Mining: The 23rd Pacific-Asia Conference*, pp.289-301, 2019.
[10] R. Li, Y. Shen and Y. Zhu, Next point-of-interest recommendation with temporal and multi-level context attention, *Proc. of the 2018 IEEE International Conference on Data Mining*, pp.1110-1115, 2018.
[11] Q. Liu, S. Wu, L. Wang and T. Tan, Predicting the next location: A recurrent model with spatial and temporal contexts, *Proc. of the 30th AAAI Conference on Artificial Intelligence*, pp.194-200, 2016.
[12] J. Li, G. Guo and X. Hu, Prediction algorithm of industry rotation based on attention LSTM model, *International Journal of Innovative Computing, Information and Control*, vol.18, no.6, pp.1969-1977, 2022.
[13] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen and Y. Rui, GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation, *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.831-840, 2014.
[14] M. Ye, P. Yin, W.-C. Lee and D.-L. Lee, Exploiting geographical influence for collaborative point-of-interest recommendation, *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.325-334, 2011.
[15] C. Song, T. Koren, P. Wang and A.-L. Barabási, Modelling the scaling properties of human mobility, *Nature Physics*, vol.6, no.10, pp.818-823, 2010.
[16] R. He, S. Liu, S. He and K. Tang, Multi-domain active learning: Literature review and comparative study, *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol.7, no.3, pp.791-804, 2023.
[17] S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, *arXiv Preprint*, arXiv: 1205.2618, 2012.
[18] S. Feng, G. Cong, B. An and Y. M. Chee, POI2Vec: Geographical latent representation for predicting future visitors, *Proc. of the 31st AAAI Conference on Artificial Intelligence*, pp.102-108, 2017.
[19] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng and X. Zhou, Where to go next: A spatio-temporal gated network for next POI recommendation, *IEEE Transactions on Knowledge and Data Engineering*, vol.34, no.5, pp.2512-2524, 2020.
[20] C. Yang, M. Sun, W. X. Zhao, Z. Liu and E. Y. Chang, A neural network approach to jointly modeling social networks and mobile trajectories, *ACM Transactions on Information Systems*, vol.35, no.4, pp.1-28, 2017.
[21] D. Lian, Q. Liu and E. Chen, Personalized ranking with importance sampling, *Proc. of the Web Conference 2020*, pp.1093-1103, 2020.
[22] W. Chen, S. Liu, Y.-S. Ong and K. Tang, Neural influence estimator: Towards real-time solutions to influence blocking maximization, *arXiv Preprint*, arXiv: 2308.14012, 2023.
[23] J.-D. Zhang and C.-Y. Chow, iGSLR: Personalized geo-social location recommendation: A kernel density estimation approach, *Proc. of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp.334-343, 2013.

[24] C. Cheng, H. Yang, I. King and M. Lyu, Fused matrix factorization with geographical and social influence in location-based social networks, *Proc. of the 26th AAAI Conference on Artificial Intelligence*, pp.17-23, 2012.

[25] B. Liu, Y. Fu, Z. Yao and H. Xiong, Learning geographical preferences for point-of-interest recommendation, *Proc. of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1043-1051, 2013.

[26] D. Lian, Y. Ge, F. Zhang, N. J. Yuan, X. Xie, T. Zhou and Y. Rui, Scalable content-aware collaborative filtering for location recommendation, *IEEE Transactions on Knowledge and Data Engineering*, vol.30, no.6, pp.1122-1135, 2018.

[27] D. Lian, K. Zheng, Y. Ge, L. Cao, E. Chen and X. Xie, GeoMF++: Scalable location recommendation via joint geographical modeling and matrix factorization, *ACM Transactions on Information Systems*, vol.36, no.3, pp.1-29, 2018.

[28] J.-D. Zhang, C.-Y. Chow and Y. Li, iGeoRec: A personalized and efficient geographical location recommendation framework, *IEEE Transactions on Services Computing*, vol.8, no.5, pp.701-714, 2014.

[29] M. Quadrana, P. Cremonesi and D. Jannach, Sequence-aware recommender systems, *ACM Computing Surveys*, vol.51, no.4, pp.1-36, 2018.

[30] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun and D. Lian, A survey on session-based recommender systems, *ACM Computing Surveys*, vol.54, no.7, pp.1-38, 2021.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in Neural Information Processing Systems*, vol.30, pp.5998-6008, 2017.

[32] S. Liu, N. Lu, C. Chen and K. Tang, Efficient combinatorial optimization for word-level adversarial textual attack, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol.30, pp.98-111, 2022.

[33] S. Liu, N. Lu, W. Hong, C. Qian and K. Tang, Effective and imperceptible adversarial textual attack via multi-objectivization, *ACM Transactions on Evolutionay Learnning and Optimization*, vol.4, no.3, DOI: 10.1145/3651166, 2024.

[34] K. Zhao, S. Liu, J. X. Yu and Y. Rong, Towards feature-free TSP solver selection: A deep learning approach, *Proc. of 2021 International Joint Conference on Neural Networks*, pp.1-8, 2021.

[35] S. Liu, K. Tang and X. Yao, Memetic search for vehicle routing with simultaneous pickup-delivery and time windows, *Swarm and Evolutionary Computation*, vol.66, 100927, 2021.

[36] S. Liu, Y. Zhang, K. Tang and X. Yao, How good is neural combinatorial optimization? A systematic evaluation on the traveling salesman problem, *IEEE Computational Intelligence Magazine*, vol.18, no.3, pp.14-28, 2023.

[37] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li and K. Tang, Disentangled contrastive learning for social recommendation, *Proc. of the 31st ACM International Conference on Information & Knowledge Management*, pp.4570-4574, 2022.

[38] W.-C. Kang and J. McAuley, Self-attentive sequential recommendation, *Proc. of 2018 IEEE International Conference on Data Mining*, pp.197-206, 2018.

[39] S. Zhang, Y. Tay, L. Yao and A. Sun, Next item recommendation with self-attention, *arXiv Preprint*, arXiv: 1808.06414, 2018.

[40] S. Liu, K. Tang and X. Yao, Automatic construction of parallel portfolios via explicit instance grouping, *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, pp.1560-1567, 2019.

[41] S. Liu, K. Tang and X. Yao, Generative adversarial construction of parallel portfolios, *IEEE Transactions on Cybernetics*, vol.52, no.2, pp.784-795, 2022.

[42] K. Tang, S. Liu, P. Yang and X. Yao, Few-shots parallel algorithm portfolio construction via co-evolution, *IEEE Transactions on Evolutionary Computation*, vol.25, no.3, pp.595-607, 2021.

[43] S. Liu, F. Peng and K. Tang, Reliable robustness evaluation via automatically constructed attack ensembles, *Proc. of the 37th AAAI Conference on Artificial Intelligence*, pp.8852-8860, 2023.

[44] J. Li, Y. Wang and J. McAuley, Time interval aware self-attention for sequential recommendation, *Proc. of the 13th International Conference on Web Search and Data Mining*, pp.322-330, 2020.

[45] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv Preprint*, arXiv: 1810.04805, 2018.

[46] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou and P. Jiang, BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer, *Proc. of the 28th ACM International Conference on Information and Knowledge Management*, pp.1441-1450, 2019.

[47] N. Lu, S. Liu, Z. Zhang, Q. Wang, H. Liu and K. Tang, Less is more: Understanding word-level textual adversarial attack via n-gram frequency descend, *arXiv Preprint*, arXiv: 2302.02568, 2023.
[48] J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie and G. Sun, xDeepFM: Combining explicit and implicit feature interactions for recommender systems, *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1754-1763, 2018.
[49] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, Y. Jin, H. Li and K. Gai, Deep interest network for click-through rate prediction, *Proc. of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1059-1068, 2018.
[50] D. Lian, Y. Wu, Y. Ge, X. Xie and E. Chen, Geography-aware sequential location recommendation, *Proc. of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.2009-2019, 2020.
[51] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, Berlin, Heidelberg, 2006.
[52] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, *arXiv Preprint*, arXiv: 1301.3781, 2013.
[53] A. Najjar and K. Mede, Trajectory-user linking is easier than you think, *Proc. of 2022 IEEE International Conference on Big Data*, pp.4936-4943, 2022.
[54] P. Kefalas, P. Symeonidis and Y. Manolopoulos, Recommendations based on a heterogeneous spatio-temporal social network, *World Wide Web*, vol.21, no.2, pp.345-371, 2018.
[55] S. Feng, L. V. Tran, G. Cong, L. Chen, J. Li and F. Li, HME: A hyperbolic metric embedding approach for next-POI recommendation, *Proc. of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.1429-1438, 2020.
[56] B. Hidasi, A. Karatzoglou, L. Baltrunas and D. Tikk, Session-based recommendations with recurrent neural networks, *arXiv Preprint*, arXiv: 1511.06939, 2015.

Author Biography

Xuan Luo received the B.Sc. degree in Digital Media Technology from Yunnan University, China, in 2013; the M.S. degree in Computer Science from University of Hong Kong, Hong Kong, in 2017.

He is currently a full-time staff of Shenzhen Insititute of Information Technology in Shenzhen, China. His main research interests in the area of computer science include intelligence system, multimedia information processing and information system optimization.

Mingqing Huang received B.Sc. degree in Automation from Hebei University of Engineering, in 2008, the M.S. degree in Software Engineering from University of Science and Technology of China, in 2011, and the Ph.D. degree in Computer Science from Shanghai University, in 2018.

He is currently a lecturer with the Shenzhen Institute of Information Technology. His major research interests include biological information, social computing, and machine learning.

Rui Lv received the M.S. degree in Electronic Science and Technology from Dalian University of Technology, Dalian, China, in 2017.

He is currently working about electronics with the Shenzhen Institute of Information Technology, Shenzhen, China. His research interest includes antenna and semiconductor.

Hui Zhao received the B.Sc. degree in Computer Science and Technology from Central South University, Changsha, China, in 2006, the M.S. degree in Computer Science and Technology from National University of Defense Technology, Changsha, China, in 2008, and Ph.D. degree in Computer Science and Technology from National University of Defense Technology, Changsha, China, in 2013.

He is currently working about cyber security with the Shenzhen Institute of Information Technology, Shenzhen, China. His research interests include cloud computing and artificial intelligence.