Counterfactual User Sequence Synthesis Augmented with Continuous Time Dynamic Preference Modeling for Sequential POI Recommendation

Lianyong Qi[1,2,3] , **Yuwen Liu**[1] , **Weiming Liu**[4] , **Shichao Pei**[5] , **Xiaolong Xu**[6*] , **Xuyun Zhang**[7] , **Yingjie Wang**[8] and **Wanchun Dou**[9*]
[1]College of Computer Science and Technology, China University of Petroleum (East China)
[2]Yunnan Key Laboratory of Service Computing
[3]School of Computer Science, Qufu Normal University
[4]College of Computer Science, Zhejiang University
[5]University of Massachusetts Boston
[6]School of Computer and Software, Nanjing University of Information Science and Technology
[7]Department of Computing, Macquarie University
[8]School of Computer and Control Engineering, Yantai University
[9]Department of Computer Science and Technology, Nanjing University
lianyongqi@upc.edu.cn, {yuwenliu97, lwming95}@gmail.com, shichao.pei@umb.edu, xlxu@nuist.edu.cn, xuyun.zhang@mq.edu.au, wangyingjie@ytu.edu.cn, douwc@nju.edu.cn

Abstract

With the proliferation of Location-based Social Networks (LBSNs), user check-in data at Points-of-Interest (POIs) has surged, offering rich insights into user preferences. However, sequential POI recommendation systems always face two pivotal challenges. A challenge lies in the difficulty of modeling time in a discrete space, which fails to accurately capture the dynamic nature of user preferences. Another challenge is the inherent sparsity and noise in continuous POI recommendation, which hinder the recommendation process. To address these challenges, we propose counterfactual user sequence synthesis with continuous time dynamic preference modeling (CussCtpm). CussCtpm innovatively combines Gated Recurrent Unit (GRU) with neural Ordinary Differential Equations (ODEs) to model user preferences in a continuous time framework. CussCtpm captures user preferences at both the POI-level and interest-level, identifying deterministic and non-deterministic preference concepts. Particularly at the interest-level, we employ GRU and neural ODEs to model users' dynamic preferences in continuous space, aiming to capture finer-grained shifts in user preferences over time. Furthermore, CussCtpm utilizes counterfactual data augmentation to generate counterfactual positive and negative user sequences. Our extensive experiments on two widely-used public datasets demonstrate that CussCtpm outperforms several advanced baseline models.

*Corresponding author

1 Introduction

With the widespread adoption of Location-based Social Networks (LBSNs), an increasing number of users are engaging in checking in for their favorite Points-of-Interest (POIs) on LBSNs. This trend, representing a significant shift in how individuals interact with their environment, has led to an exponential increase in user check-in data. Such data not only offers insights into users' preferences and behaviors but also provides a unique opportunity to enhance user engagement through tailored experiences [Islam *et al.*, 2022]. While there have been many works in the field of POI recommendations, early efforts [Horozov *et al.*, 2006; Devarajan *et al.*, 2019] primarily focus on the interactions between users and POIs without considering the temporal effect of user check-ins. Temporal information undeniably plays a crucial role in influencing user check-ins. Recent advances [Lim *et al.*, 2022; Lim *et al.*, 2020] have explored sequential POI recommendations considering the impact of time. These advances have attempted to predict users' subsequent check-ins based on their historical interactions with POIs.

Despite these advancements, sequential POI recommendation still faces numerous challenges. An important challenge (C1) lies in the difficulty of modeling time in a discrete space, which fails to accurately capture the dynamic nature of user preferences. Although many studies [Rahmani *et al.*, 2022; Kumar *et al.*, 2024] regard time as an important factor affecting user check-in and consider time factors in their studies, these studies have all modeled time in a discrete space. However, user preferences towards certain POIs change over time. For example, a user's preference for a supermarket may decrease with increased waiting time. Time modeling in discrete space cannot reflect the fine-grained continuous preference changes of users, but modeling in continuous space can solve this problem. This represents a unique and significant issue in sequential POI recommendation, which is different from other recommendation systems. Another persistent challenge (C2) is the sparsity issue inherent in sequential POI recommendations. The vast number of users and POIs inevitably leads to sparse interactions, which is a fundamental issue that cannot be overlooked [Liu *et al.*, 2023b]. The additional challenge (C3) is the noise in POI recommendations, arising from extrinsic factors, further complicates the scenario. Users might check in at POIs that do not genuinely reflect their preferences, influenced by external factors like restaurant promotions or social events. Some works alleviate the above challenges by using adaptive denoising techniques [Wang *et al.*, 2021a] and contrastive learning techniques [Gu *et al.*, 2022]. However, the sparsity and noise has been the challenge that is difficult to ignore in recommendations.

We propose an approach that augments **C**ounterfactual **U**ser **S**equence **S**ynthesis with **C**ontinuous **T**ime Dynamic **P**reference **M**odeling for sequential POI Recommendation, abbreviated as **CussCtpm**. To tackle the challenge (C1) of modeling dynamic user preferences and capture the precise user preferences in a continuous space, we utilize GRU to model the evolving nature of user preferences and then employ neural Ordinary Differential Equations (ODEs) [Chen *et al.*, 2018] to transition the modeling of user time preferences from a discrete space to a continuous space. To mitigate the issues of sparsity (C2) and noise (C3), we use counterfactual data to enhance the learning of user sequences. Specifically, we define two concepts: the deterministic preference, which represents a user's inherent likes, and the non-deterministic preference, which depicts less favored interests. By substituting the fine-grained POI-level or abstract interest-level of these deterministic and non-deterministic preferences, we achieve counterfactual transformations. Subsequently, by using contrastive learning, we capture the representations that more accurately reflect the user's true preferences. Our primary contributions are summarized as follows:

- We propose the transition of dynamic preference modeling from a discrete to a continuous space, utilizing ODEs to capture the continuous evolution of user preferences.
- We leverage counterfactual data to alleviate the negative impacts of data sparsity and noise, refining the representation of user preferences by substituting deterministic and non-deterministic preferences.
- We conduct extensive experiments on two public POI datasets, demonstrating the superiority of CussCtpm.

2 Related Work

2.1 Sequential POI Recommendation

Early works use collaborative filtering method [Cai *et al.*, 2013] to make POI recommendations based on user location information [Zhao *et al.*, 2020] or POI category data [Rahmani *et al.*, 2019]. However, these methods overlook the temporal aspect of user check-ins. Subsequent research begins to capture the time effects within user check-in sequences using Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] and Gated Recurrent Unit (GRU) [Chung *et al.*, 2014; Qi *et al.*, 2022]. Another effort [Manotumruksa *et al.*, 2018] utilizes an attention mechanism combined with GRU to capture the dynamic preferences of users by leveraging feedback sequences and associated contextual information. In addition, Liu et al. [Liu *et al.*, 2022] apply graph convolutional models to POI recommendations, thereby extracting user preference embedded within higher-order connectivity. However, despite these advances in incorporating time information into recommendation systems, a key limitation still exists: discrete spatial modeling of time effects. This limitation hinders the extraction of fine-grained changes in user preferences, which is also the challenge that our research aims to address.

The exploration of user dynamic preferences has been a key area of sequential POI recommendations. To enhance the recommendation performance, many advances [Tan *et al.*, 2021; Liu *et al.*, 2023a; Bian *et al.*, 2023] have conducted extensive research on capturing user dynamic preferences. A recent effort [Passino *et al.*, 2021] proposes a dynamic model for user preferences towards item categories. This model can estimate the probability of transitions between item categories over time, thereby predicting user preferences. Furthermore, the use of ODEs for modeling user preferences in a continuous space enables the fine-grained capture of shifts in user preferences. A study [Tan *et al.*, 2021] leverages meta-learning enhanced neural ODEs to extract dynamic user preferences. Similarly, a work [Bian *et al.*, 2023] defines the ODEs for the history graph and context graph to track the dynamic changes in user preferences. Therefore, we can use ODEs to model the continuous preferences of users, which can capture the authentic changes in their preferences and provide better recommendations.

2.2 Counterfactual Data Augmentation

Counterfactual data augmentation is commonly used to alleviate data scarcity issues and has been successfully applied in various fields such as computer vision [Singla *et al.*, 2023]. However, its application in recommendation systems is underexplored. A recent study [Yang *et al.*, 2021] applies counterfactual data augmentation in Top-N recommendation systems. Similarly, subsequent research [Saito and Joachims, 2021; Liu *et al.*, 2020] predominantly aims to reduce bias in learning-level problems. To alleviate the impact of adverse data in the sequential recommendation, an attempt [Wang *et al.*, 2021b] employs sampler and anchor models for heuristic intervention during model training. A work [Zhang *et al.*, 2021] applies counterfactual data augmentation in the recommendation field, utilizing counterfactual user sequence synthesis to alleviate the noise and sparsity issues.

3 Preliminaries

3.1 Problem Definition

Let $\mathrm{U} = \{u_1, u_2, ..., u_{|\mathrm{U}|}\}$ denote the set of users, and $\mathrm{P} = \{p_1, p_2, ..., p_{|\mathrm{P}|}\}$ denote the set of POIs, where $|\mathrm{U}|$ and $|\mathrm{P}|$ represent the total number of users and POIs, respectively. Each user $u \in \mathrm{U}$ is associated with a sequence of interactions with POIs over time. We denote the sequence of historical interactions for user u as $S_u = \{p_1, p_2, ..., p_{t-1}\}$, where

$p_{t-1} \in \mathrm{P}$ represents the POI interacted with, and t denotes the corresponding timestamp of interaction.

Given the check-in sequence of a user up to time $t-1$, our goal is to predict the POI that he/she probably visit at time t. Formally, the prediction target for user u at time t is denoted as $g_u = p_t$, and the prediction task is formulated in (1):

$$\hat{p}_t = \arg max_{p \in \mathrm{P}} \Pr(p|S_u, t; \epsilon), \tag{1}$$

where $\hat{p}_t$ is the predicted POI, $\Pr(p|S_u, t; \epsilon)$ is the probability of visiting POI p given the historical sequence S_u and time t, parameterized by ϵ.

3.2 Counterfactual Transformation

In the counterfactual transformation process, we apply artificial interventions to change certain preferences of users to observe variations in their behavior. This process is achieved by replacing deterministic and non-deterministic preferences of users. We define the two concepts as follows:

Deterministic Preference. Deterministic preference refers to actions that are crucial to users' interest representation. These preferences reflect the core interests of the user and are not arbitrarily replaceable.

Non-Deterministic Preference. Non-deterministic preference represents actions that are irrelevant to the user's core interests and may even constitute noise. They are less important for representing users' true interests.

3.3 Neural ODE for Modeling Continuous Dynamics

Neural ODE [Chen *et al.*, 2018] offers a principled framework for modeling the evolution of systems over continuous time. In user preference modeling, ODE enables us to capture the dynamic nature of user behavior as it unfolds through time. An ODE is a mathematical equation that describes the relationship between a function and its derivatives, representing the rate of change of the function with respect to an independent variable, typically time in our task. Formally, an ODE for a function $\mathbf{h}(t)$ can be expressed as:

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t; \Theta), \tag{2}$$

where $\mathbf{h}(t)$ denotes the state of the system at time t, $\frac{d\mathbf{h}(t)}{dt}$ is the derivative of $\mathbf{h}(t)$ with respect to time, and f is a function parameterized by Θ.

Incorporating ODE into our model necessitates the accurate integration of the user preference state over time. Given an initial state $\mathbf{h}(t_0)$, we can derive the state at any later time T using a black-box differential equation solver defined as:

$$\mathbf{h}(T) = \mathbf{h}(t_0) + \int_{t_0}^{T} f(\mathbf{h}(t), t; \Theta) dt, \tag{3}$$

where the integral computes the accumulation of changes in preferences from the initial time t_0 to any given time T.

The use of an ODE solver in our model is formalized as:

$$\mathbf{h}_{\text{continuous}}(T) = \text{odeint}(f, \mathbf{h}(t_0), T, \Theta), \tag{4}$$

where $\text{odeint}(\cdot)$ denotes the black-box ODE solver [Ahnert and Mulansky, 2011], and $\mathbf{h}_{\text{continuous}}(T)$ is the state of user preferences at time T. The solver provides a differentiable operation that is integrated with the automatic differentiation capabilities of deep learning frameworks, allowing for end-to-end learning of the parameters Θ through gradient descent.

4 The Proposed Method

The proposed CussCtpm framework is divided into two main components: continuous dynamic preference capture and concept identification, counterfactual behavior representation. The framework is schematically shown in Figure 1.

4.1 Continuous Dynamic Preference Capture and Concept Identification

POI-level Concepts

Inspired by a recent work [Zhang *et al.*, 2021], we consider each POI within a behavior sequence as an independent POI-level concept, with each POI possessing its unique fine-grained characteristics. Next, in order to encode the user's behavioral sequence, we utilize $\mathbf{B} = p^e_\epsilon(S_{u,1:t})$ as the encoded representation of the concept sequence, where $\mathbf{S} = p^e_\epsilon(S_{u,1:t})$ denotes the representations of the behavior sequence and $\mathbf{S} \in \mathbb{R}^{t \times d}$ where d denotes the dimension of each representation. Here, $p^e_\epsilon(\cdot)$ is the POI encoder, and $(S_{u,1:t})$ represents the complete check-in sequence of the user. The concept score at the POI level, reflecting the importance of a concept in representing user interests, is computed as follows:

$$p_n^{\text{poi}} = sim(\mathbf{b}_n \cdot \mathbf{g}_u), \tag{5}$$

where $\mathbf{b}_n$ denotes the concept vector for the n-th POI, and $\mathbf{g}_u$ is the embedding of the target POI for user u, $sim(\cdot)$ is a similarity function calculated by inner product.

Interest-level Concepts

Solely utilizing POI-level representations may lead to redundancy in concepts due to the possibility of shared semantics across POIs. To overcome this, we introduce interest-level concepts by capturing the continuous dynamic preferences of users. This process involves identifying deterministic and non-deterministic preference concepts at a more abstract level. We use GRU to capture short-term and long-term preferences containing user interests. This is mainly achieved through the gate structure in GRU, defined by $\boldsymbol{h}_t = \text{GRU}(\boldsymbol{h}_{t-1}, \boldsymbol{p}_t)$. The details are defined as:

$$\begin{aligned}
\boldsymbol{r}_t &= \sigma\left(\boldsymbol{W}_r \boldsymbol{p}_t + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \\
\boldsymbol{z}_t &= \sigma\left(\boldsymbol{W}_z \boldsymbol{p}_t + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \\
\tilde{\boldsymbol{h}}_t &= \tanh\left(\boldsymbol{W}_h \boldsymbol{p}_t + \boldsymbol{U}_h \left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right) \\
\boldsymbol{h}_t &= \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - z_t) \odot \tilde{\boldsymbol{h}}_t
\end{aligned} \tag{6}$$

where $\boldsymbol{r}_t$ and $\boldsymbol{z}_t$ correspond to the reset and update gates, respectively, while $\tilde{\boldsymbol{h}}_t$ represents the candidate activation, and $\boldsymbol{h}_t$ represents the final state of GRU. $\boldsymbol{p}_t$ is the embedding of a POI at the present timestep t, and $\boldsymbol{h}_{t-1}$ is the hidden state of the preceding timestep $t-1$. The parameters $\{\boldsymbol{W}_r, \boldsymbol{U}_r, \boldsymbol{W}_z, \boldsymbol{U}_z, \boldsymbol{W}_h, \boldsymbol{U}_h\}$ are the trainable matrices, $\{\boldsymbol{b}_r, \boldsymbol{b}_z, \boldsymbol{b}_h\}$ are biases. The sigmoid activation function is symbolized by σ, and the Hadamard product is denoted by $\odot$.

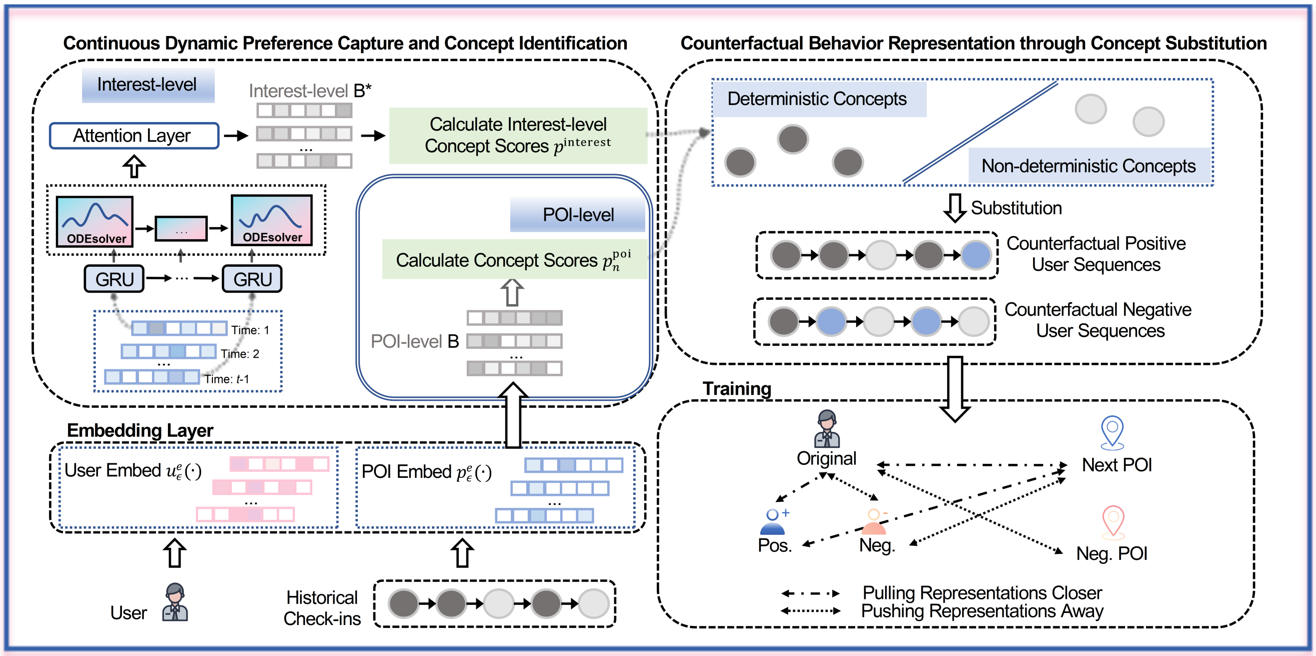


Figure 1: Overview of the CussCtpm framework

Other works [Manotumruksa *et al.*, 2018; Liu *et al.*, 2021] utilize GRU and combine attention mechanisms when capturing users' dynamic preferences. Although such methods can capture user preferences, they still model user preferences in a discrete space. Modeling in discrete space is unable to capture fine-grained user preferences. Here, we use neural ODEs to map users' dynamic preferences from discrete space to continuous space, as shown in Figure 2 for a specific example. Here, it is assumed that the user's preference for POIs is relatively constant in discrete space, whereas the user's preference for POIs alternates between decreasing/increasing in continuous space. Our goal is to model user preferences within a continuous domain, hence we evolve from the discrete time step at $t-1$ in our original domain, projecting it onto a continuous span represented by $t-\Delta t$ within our destination domain. Through this transition, the difference of the GRU's outputs, denoted by $\Delta \boldsymbol{h}_t$, are utilized to formulate a differential equation as follows:

$$\begin{aligned}\Delta \boldsymbol{h}_t &= \boldsymbol{h}_t - \boldsymbol{h}_{t-\Delta t}\\ &= \boldsymbol{z}_t \odot \boldsymbol{h}_{t-\Delta t} + (1-\boldsymbol{z}_t)\odot \tilde{\boldsymbol{h}}_t - \boldsymbol{h}_{t-\Delta t}\\ &= (\boldsymbol{z}_t - 1)\odot \boldsymbol{h}_{t-\Delta t} + (1-\boldsymbol{z}_t)\odot \tilde{\boldsymbol{h}}_t\\ &= (1-\boldsymbol{z}_t)\odot\left(\tilde{\boldsymbol{h}}_t - \boldsymbol{h}_{t-\Delta t}\right)\end{aligned} \tag{7}$$

This difference equation naturally leads to the following ODE for $\boldsymbol{h}(t)$ when $\Delta t \to 0$:

$$\frac{d\boldsymbol{h}(t)}{dt} = (1-\boldsymbol{z}(t))\odot\left(\tilde{\boldsymbol{h}}(t) - \boldsymbol{h}(t)\right) \tag{8}$$

Through GRU and the subsequent transformation into continuous time representations via ODE, our model employs an

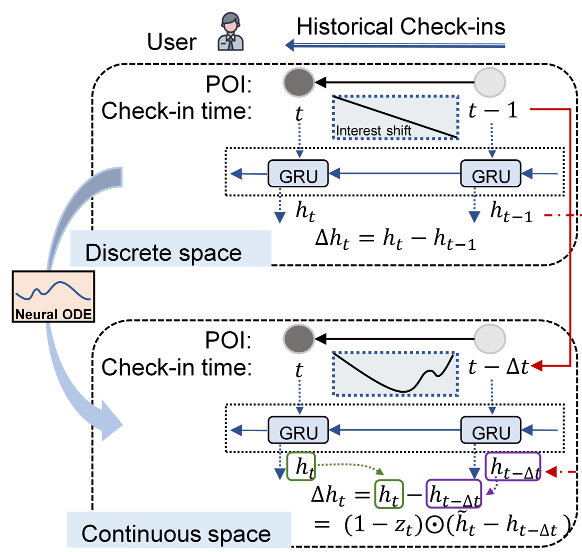


Figure 2: Mapping user preferences from discrete space to continuous space through neural ODE: an example

attention mechanism to further refine user interest representations. The interest-level concepts are then extracted to weigh the importance of different points in the sequence as follows:

$$\alpha = \mathrm{softmax}(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{H}^\top))^\top, \tag{9}$$

where $\mathbf{H}$ is the vectorial representations of the behavior sequence containing dynamic preferences, $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable parameters, α is the attention score by an additive model, and $\tanh(\cdot)$ represents hyperbolic tangent function. Then we can obtain the concept sequence as follows:

$$\mathbf{B}^* = \alpha^\top \mathbf{H} \tag{10}$$

Given that the interest-level concepts and target POI are not naturally embedded in the same space, we calculate the concept score using a weighted sum at the POI-level.

$$p^{\text{interest}} = \alpha^\top sim(\mathbf{H} \cdot \mathbf{g}_u) \tag{11}$$

Finally, we obtain the POI-level concepts and the interest-level concepts, respectively. Subsequently, it is necessary to distinguish the importance of these concepts. Consistent with the approach in [Zhang *et al.*, 2021], we categorize the top 50% of the scores from the POI-level ratings p_n^{poi} and interest-level ratings p^{interest} as deterministic preferences, while the rest are classified as non-deterministic preferences. This selection ensures a balanced proportion between deterministic and non-deterministic preferences in our analysis.

4.2 Counterfactual Behavior Representation

If we replace the deterministic preferences in the user sequence, the core interests and preferred POIs of the user will be changed by which the sequence becomes a counterfactual negative user sequence. On the contrary, if the non-deterministic preferences in the user sequence are replaced, the new sequence will still be semantically similar to the original sequence, and the new sequence will become a counterfactual positive user sequence. Here, we set a fixed replacement ratio γ as the proportion of deterministic and non-deterministic concepts that are substituted to create these counterfactual sequences. To facilitate the concept substitution process, we employ a First-In-First-Out (FIFO) queue as a concept memory bank at each level. This queue serves as a repository for concepts extracted from the current mini-batch, which are enqueued for subsequent use. By employing this FIFO queue mechanism, we effectively maintain the overall sequence length and the relative positions of the remaining concepts, which is critical for preserving the structural integrity of the user sequence.

Inspired by [Zhang *et al.*, 2021; Clark *et al.*, 2020], we consider the contrast between counterfactual sequences and observation. A robust user representation should exhibit minimal sensitivity to non-deterministic preference concepts, while an accurate representation should rely more on deterministic preference concepts. To achieve this, we use triplet margin loss to measure the relative similarity between the representations derived from the original and counterfactual sequences. The loss function for this contrast is given by:

$$L_{c_1} = \sum_{y=1}^{Y}\sum_{x=1}^{X} \max\left\{d\left(\mathbf{u}, \mathbf{u}_y^+\right) - d\left(\mathbf{u}, \mathbf{u}_x^-\right) + \Delta_{c_1}, 0\right\} \tag{12}$$

where $\mathbf{u}, \mathbf{u}_y^+, \mathbf{u}_x^-$ represent the true representation of the user, counterfactual positive user representation, and counterfactual negative user representation from counterfactual POI-level concept sequences, respectively. Additionally, x and y denote the number of generated counterfactual negative users and counterfactual positive users, respectively. We choose the L2 distance as the distance function $d(\cdot)$, and empirically set the margin $\Delta_{c_1} = 1$.

The aforementioned goal focuses exclusively on user representation. We also incorporate the target POI which contributes to refining user representation learning.

$$L_{c_2} = \sum_{y=1}^{Y} 1 - \hat{\mathbf{u}}_y^+ \cdot \hat{\mathbf{p}} + \sum_{x=1}^{X} \max\left(0, \hat{\mathbf{u}}_x^- \cdot \hat{\mathbf{p}} - \Delta_{c_2}\right) \tag{13}$$

Here, $\hat{\mathbf{p}}$ denote the target POI, the margin Δ_{c_2} is set to 0.5.

4.3 Optimization

To enhance the accuracy and robustness of user representations, our model employs contrastive learning. This approach is pivotal for training the model, as it emphasizes the relationships between different types of users. We first define the primary loss function for our model training:

$$\begin{aligned} L_p &= \arg\min_\theta \frac{1}{|\mathcal{V}|} \sum_{(S_u, p_t)\in\mathcal{V}} -\log o_\theta(p_t \mid S_u), \\ \text{where } o_\theta(p_t \mid S_u) &= \frac{\exp \psi_\theta(p_t, S_u)}{\sum_{p_t' \in |P|} \exp \psi_\theta\left(S_u, p_t'\right)} \end{aligned} \tag{14}$$

Here, $\psi_\theta(\cdot)$ denotes cosine similarity, p_t' denotes all possible POIs that may be accessed, $\mathcal{V} = \{(S_{u_m}, g_{u_m}) | u_m \in \mathrm{U}\}$ denotes the user's historical interaction behaviors, and θ denotes the corresponding trainable parameters.

Combining the above components, the overall loss function for our model is given by:

$$L_{total} = L_p + \lambda \mathcal{L}_{c_1} + \eta \mathcal{L}_{c_2}, \tag{15}$$

where λ and η are weighting parameters that balance the contributions of contrastive objectives to total loss.

5 Experiments

In this section, we have conducted extensive experiments on real-world datasets, and analyzed the performance of the proposed CussCtpm by addressing the following questions:

- **RQ1:** Can our proposed CussCtpm outperform the baselines for sequential POI recommendation?
- **RQ2:** How does the continuous time dynamic preference modeling in CussCtpm affect model performance?
- **RQ3:** How to demonstrate the effectiveness of synthesizing counterfactual user sequences?
- **RQ4:** How is the performance of CussCtpm affected by different parameter settings?

5.1 Experimental Settings

Datasets and Metrics. We utilize real-world check-in datasets: NYC and TKY collected from Foursquare. These datasets [Yang *et al.*, 2013] encompass user check-in details. We consider POIs with less than 5 check-ins as outliers and delete them. For evaluation metrics, we employ Mean Reciprocal Rank (MRR) and Hit Rate (HIT). MRR can quantify the accuracy of the top-ranked recommendations and emphasize the importance of placing truly relevant POIs at the forefront. HIT can evaluate the overall effectiveness of our POI recommendation system, reflecting how frequently our proposed POIs align with users' genuine preferences. Recommendation number Top-K candidates are 5, 10, 20.

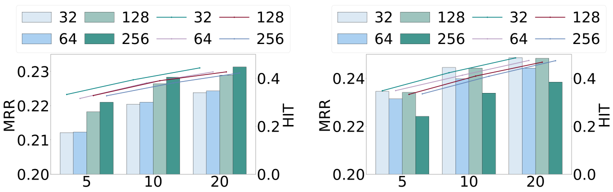
Baselines. Our experimental study compares CussCtpm with a series of methods, including traditional methods like **BPR** [Rendle *et al.*, 2009], contemporary GNN-based methods such as **LESSR** [Chen and Wong, 2020], **SR-GNN** [Wu *et al.*, 2019], **GCE-GNN** [Wang *et al.*, 2020], **NISER** [Gupta *et al.*, 2019], simplified GCN-based methods such as **NGCF** [Wang *et al.*, 2019], **PriGCN** [Liu *et al.*, 2023c], and the time-aware method such as **ITGCN** [Liu *et al.*, 2022].

5.2 Results and Discussion (RQ1)

Our experiment compares CussCtpm with a series of methods. The comparison results are summarized in Table 1. From Table 1, we can observe that our method is overall superior to other methods and achieves good performance improvement. The traditional method BPR is greatly affected by the dataset and performs poorly on the NYC dataset, but ensures good performance on the TKY dataset. This may be because BPR does have personalized sorting capabilities, but its universality needs to be improved. The GNN-based methods, capable of capturing high-order user-POI connectivity, are yet hindered by data sparsity and noise, as evidenced by GCE-GNN's performance on NYC. Compared with simplified GCN-based methods, these models omit some unnecessary structures in the graph, such as activation functions and transformation matrices. These methods have improved training speed, but have not overcome the effects of sparsity and noise. Considering the influence of time, ITGCN models can capture the dynamic preferences of users in discrete space. However, user preferences are constantly changing over time, while ITGCN can not consider this influencing factor. In contrast, CussCtpm excels in addressing these challenges by integrating continuous-time dynamic preference modeling and counterfactual user sequence synthesis. This approach not only captures the evolving nature of user preferences in a more granular manner but also effectively mitigates issues arising from sparse data and extrinsic noise.

5.3 Impact of Capturing User Preferences Dynamically (RQ2)

We evaluate the impact of continuous time dynamic preference modeling on the performance of CussCtpm. We create a variant of our model, CussCtpm-t, which omits the integration of GRU with Neural ODE for capturing the dynamic preferences of users, relying solely on attention mechanisms for preference capture. The comparative analysis of CussCtpm and its variant CussCtpm-t is summarized in Table 2. The results indicate a noticeable decline in recommendation performance when the dynamic preference capture component is excluded. Specifically, CussCtpm-t exhibits lower MRR and HIT scores compared to the complete CussCtpm model. These findings underscore the significance of capturing user preferences dynamically, leveraging the synergy between GRU and neural ODE.



(a) NYC (b) TKY

Figure 3: Performance w.r.t. different embedding dimensions

5.4 Impact of Aggregation Strategy (RQ3)

We experiment with three variations of our model: CussCtpm-n (without counterfactual negative sequences), CussCtpm-p (without counterfactual positive sequences), and CussCtpm-np (without any counterfactual sequences). Table 2 summarizes the performance of these variants compared to the complete CussCtpm model. The experimental results highlight a significant decline in model performance with the omission of both counterfactual positive and negative user sequences in CussCtpm-np. CussCtpm-n, which maintains counterfactual positive user sequences, marginally outperforms CussCtpm-p, indicating a slightly greater impact of counterfactual positive sequences on the model's performance.

5.5 Parameter Sensitivity Analyses (RQ4)

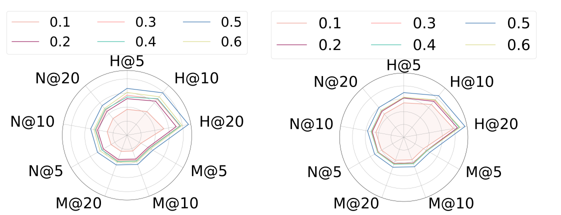
Embedding Dimension. We explore various options for the embedding dimension e, specifically: 32, 64, 128, and 256. The results are summarized in Figure 3. The model's MRR shows an increase on NYC but a decrease on TKY. This phenomenon could be attributed to the fact that an embedding dimension of 32 already offers sufficient expressive power for this scenario. Hence, merely increasing the embedding dimension may lead to more computational cost and potentially cause the feature representation overly dispersed.

Replacement Ratio. Our method generates counterfactual negative/positive users by replacing deterministic/non-deterministic preferences in user sequences. Consequently, the choice of replacement ratio γ is a crucial factor affecting model performance. We have tested several values of γ: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, and then conducted extensive experiments. The results are summarized in Figure 4. Through observation, it is evident that both in the NYC and TKY datasets, the model's performance gradually improves as the γ value increases. However, there is a noticeable decline in performance when γ shifts from 0.5 to 0.6. This indicates that the model achieves optimal performance at a replacement ratio of 0.5. A higher γ introduces excessive noise, while a lower value leads to suboptimal counterfactual learning capabilities.

The Number of Interest-level Concepts. We capture the dynamic preferences of users and generate interest-level concepts. We evaluated three different quantities: 10, 20, and 30, and summarized the results in Figure 5. Our observations indicate that with a lower number of interest-level concepts, the granularity of these concepts is coarser. As the number increases, the concepts become more fine-grained. However,

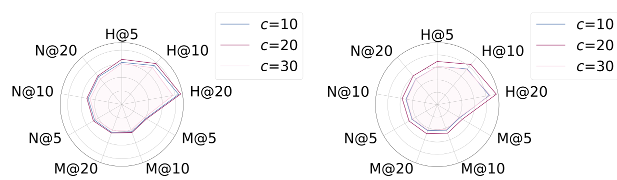
Datasets	NYC						TKY					
Metric	M@5	M@10	M@20	H@5	H@10	H@20	M@5	M@10	M@20	H@5	H@10	H@20
BPR	0.0869	0.0949	0.0983	0.1493	0.2090	0.2637	0.2243	0.2316	0.2360	0.3179	0.3744	0.4385
LESSR	0.0686	0.0701	0.0712	0.0862	0.0981	0.1190	0.1547	0.1620	0.1659	0.2305	0.2850	0.3408
SR-GNN	0.0380	0.0385	0.0388	0.0599	0.0645	0.0737	0.0754	0.0773	0.0789	0.1068	0.1198	0.1438
GCE-GNN	0.1612	0.1642	0.1692	0.2672	0.2903	0.3594	0.1551	0.1608	0.1641	0.2265	0.2723	0.3159
NISER	0.0357	0.0364	0.0364	0.0506	0.0553	0.0691	0.0505	0.0522	0.0531	0.0740	0.0980	0.1067
NGCF	0.0539	0.0587	0.0615	0.0995	0.1293	0.1741	0.1788	0.1819	0.1850	0.2538	0.2769	0.3205
PriGCN	0.2027	0.2031	0.2103	0.3521	0.3777	0.3893	0.2122	0.2200	0.2236	0.3336	0.3847	0.3951
ITGCN	0.1934	0.1989	0.2024	0.3634	0.3689	0.3705	0.2288	0.2301	0.2358	0.3354	0.3937	0.4029
CussCtpm-t	0.2134	0.2159	0.2188	0.3721	0.4105	0.4377	0.2153	0.2204	0.2298	0.3024	0.3561	0.3961
CussCtpm-n	0.2095	0.2132	0.2165	0.3694	0.4051	0.4247	0.2147	0.2159	0.2246	0.2988	0.3530	0.3897
CussCtpm-p	0.2088	0.2124	0.2134	0.3691	0.4012	0.4233	0.2139	0.2160	0.2229	0.2967	0.3529	0.3876
CussCtpm-np	0.1937	0.2031	0.2059	0.3588	0.3982	0.4200	0.2123	0.2087	0.2149	0.2900	0.3504	0.3834
CussCtpm	**0.2204**	**0.2238**	**0.2253**	**0.3934**	**0.4432**	**0.4931**	**0.2345**	**0.2445**	**0.2486**	**0.3480**	**0.4230**	**0.4850**
Improv.	8.73%	10.19%	7.13%	8.26%	17.34%	26.66%	2.49%	5.57%	5.34%	3.76%	7.44%	10.60%

Table 1: Model performance. The best performing baseline and best performer in each row are underlined and boldfaced, respectively



(a) NYC (b) TKY

Figure 4: Performance w.r.t. different replacement ratio



(a) NYC (b) TKY

Figure 5: Performance w.r.t. different interest-level concepts

x/y	Metric	NYC	TKY	NYC	TKY	x/y
1/1	H@5	0.2825	0.3101	0.2825	0.3101	1/1
	H@10	0.3333	0.3890	0.3333	0.3890	
	H@20	0.3915	0.4509	0.3915	0.4509	
	M@5	0.1834	0.2073	0.1834	0.2073	
	M@10	0.1902	0.2178	0.1902	0.2178	
	M@20	0.1942	0.2220	0.1942	0.2220	
1/2	H@5	0.2918	0.3188	0.2798	0.3083	2/1
	H@10	0.3509	0.3851	0.3416	0.3764	
	H@20	0.3980	0.4466	0.3869	0.4392	
	M@5	0.1972	0.2133	0.1779	0.2098	
	M@10	0.2048	0.2221	0.1865	0.2191	
	M@20	0.2081	0.2264	0.1896	0.2236	
1/4	H@5	0.2724	0.3210	0.3324	0.3480	4/1
	H@10	0.3343	0.3956	0.3934	0.4230	
	H@20	0.3869	0.4514	0.4432	0.4850	
	M@5	0.1886	0.2180	0.2121	0.2345	
	M@10	0.1970	0.2282	0.2204	0.2445	
	M@20	0.2007	0.2321	0.2238	0.2486	

Table 2: Performance analysis on the number of counterfactual negative/positive user representations

an excess of concepts can lead to increased noise. Therefore, the optimal number of concepts we have determined is 20.

The Number of Negative/Positive User Representations. We investigate the impact of varying the number of counterfactual negative and positive users on our model's performance. The results are summarized in Table 2. Initially, we set the count for both negative and positive users to 1. Then, we incrementally increase the number of counterfactual negative users (i.e., x) and counterfactual positive users (i.e., y), to observe their respective impacts. From Table 2, moderately increasing the quantity of negative users is beneficial. However, an excessive number of positive users can introduce unwanted noise. Therefore, we have determined the optimal configuration for our model to be $x = 4$ and $y = 1$.

6 Conclusion

In conclusion, our work presents CussCtpm, a novel approach for continuous POI recommendation in LBSNs. CussCtpm effectively addresses the critical challenges of modeling user preferences in a continuous temporal space and mitigating data sparsity. By integrating GRU with neural ODEs, our model captures nuanced, evolving user preferences at both POI and interest levels. The use of counterfactual data augmentation for generating positive and negative user sequences, combined with contrastive learning, significantly enhances the accuracy of user representations and overall recommendation performance.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant (No. 92267104, No. 62372242, No. 62272405), Natural Science Foundation of Shandong Province (No. ZR2023MF007), Foundation of Yunnan Key Laboratory of Service Computing (No. YNSC23103), Dou Wanchun Expert Workstation of Yunnan Province (No.202105AF150013). Dr Xuyun Zhang is the recipient of an ARC DECRA (project No. DE210101458) funded by the Australian Government.

References

[Ahnert and Mulansky, 2011] Karsten Ahnert and Mario Mulansky. Odeint–solving ordinary differential equations in c++. In *AIP Conference Proceedings*, volume 1389, pages 1586–1589. American Institute of Physics, 2011.

[Bian *et al.*, 2023] Qingtian Bian, Jiaxing Xu, Hui Fang, and Yiping Ke. Cpmr: Context-aware incremental sequential recommendation with pseudo-multi-task learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 120–130, 2023.

[Cai *et al.*, 2013] Yi Cai, Ho-fung Leung, Qing Li, Huaqing Min, Jie Tang, and Juanzi Li. Typicality-based collaborative filtering recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):766–779, 2013.

[Chen and Wong, 2020] Tianwen Chen and Raymond Chi-Wing Wong. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1172–1180, 2020.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[Chung *et al.*, 2014] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[Clark *et al.*, 2020] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *ICLR*, pages 26–30, 2020.

[Devarajan *et al.*, 2019] Malathi Devarajan, N Sabiyath Fatima, Subramaniyaswamy Vairavasundaram, and Logesh Ravi. Swarm intelligence clustering ensemble based point of interest recommendation for social cyber-physical systems. *Journal of Intelligent & Fuzzy Systems*, 36(5):4349–4360, 2019.

[Gu *et al.*, 2022] Shuyun Gu, Xiao Wang, Chuan Shi, and Ding Xiao. Self-supervised graph neural networks for multi-behavior recommendation. In *IJCAI*, pages 2052–2058, 2022.

[Gupta *et al.*, 2019] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M. Shroff. Niser: Normalized item and session representations to handle popularity bias. *arXiv: Information Retrieval*, 2019.

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

[Horozov *et al.*, 2006] Tzvetan Horozov, Nitya Narasimhan, and Venu Vasudevan. Using location for personalized poi recommendations in mobile environments. In *International Symposium on Applications and the Internet (SAINT'06)*, pages 124–129. IEEE Computer Society, 2006.

[Islam *et al.*, 2022] Md Ashraful Islam, Mir Mahathir Mohammad, Sarkar Snigdha Sarathi Das, and Mohammed Eunus Ali. A survey on deep learning based point-of-interest (poi) recommendations. *Neurocomputing*, 472:306–325, 2022.

[Kumar *et al.*, 2024] Akshi Kumar, Deepak Kumar Jain, Abhishek Mallik, and Sanjay Kumar. Modified node2vec and attention based fusion framework for next poi recommendation. *Information Fusion*, 101:101998, 2024.

[Lim *et al.*, 2020] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Jagannadan Varadarajan. Stp-udgat: Spatial-temporal-preference user dimensional graph attention network for next poi recommendation. In *CIKM*, pages 845–854, 2020.

[Lim *et al.*, 2022] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Yong Liang Goh, Renrong Weng, and Rui Tan. Hierarchical multi-task graph recurrent network for next poi recommendation. In *SIGIR*, pages 1133–1143, 2022.

[Liu *et al.*, 2020] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 831–840, 2020.

[Liu *et al.*, 2021] Yuwen Liu, Aixiang Pei, Fan Wang, Yihong Yang, Xuyun Zhang, Hao Wang, Hongning Dai, Lianyong Qi, and Rui Ma. An attention-based category-aware gru model for the next poi recommendation. *International Journal of Intelligent Systems*, 36(7):3174–3189, 2021.

[Liu *et al.*, 2022] Yuwen Liu, Huiping Wu, Khosro Rezaee, Mohammad R Khosravi, Osamah Ibrahim Khalaf, Arif Ali Khan, Dharavath Ramesh, and Lianyong Qi. Interaction-enhanced and time-aware graph convolutional network for successive point-of-interest recommendation in traveling enterprises. *IEEE Transactions on Industrial Informatics*, 19(1):635–643, 2022.

[Liu *et al.*, 2023a] Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiajie Su, Xinting Liao, Mengling Hu, and Yanchao Tan. Joint internal multi-interest exploration and external domain alignment for cross domain sequential recommendation. In *WWW*, pages 383–394, 2023.

[Liu *et al.*, 2023b] Weiming Liu, Xiaolin Zheng, Jiajie Su, Longfei Zheng, Chaochao Chen, and Mengling Hu. Contrastive proxy kernel stein path alignment for cross-domain cold-start recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11216–11230, 2023.

[Liu *et al.*, 2023c] Yuwen Liu, Xiaokang Zhou, Huaizhen Kou, Yawu Zhao, Xiaolong Xu, Xuyun Zhang, and Lianyong Qi. Privacy-preserving point-of-interest recommendation based on simplified graph convolutional network for geological traveling. *ACM Trans. Intell. Syst. Technol.*, 2023.

[Manotumruksa *et al.*, 2018] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. A contextual attention recurrent architecture for context-aware venue recommendation. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 555–564, 2018.

[Passino *et al.*, 2021] Francesco Sanna Passino, Lucas Maystre, Dmitrii Moor, Ashton Anderson, and Mounia Lalmas. Where to next? a dynamic model of user preferences. *WWW*, 21:19–23, 2021.

[Qi *et al.*, 2022] Lianyong Qi, Yuwen Liu, Yulan Zhang, Xiaolong Xu, Muhammad Bilal, and Houbing Song. Privacy-aware point-of-interest category recommendation in internet of things. *IEEE Internet of Things Journal*, 9(21):21398–21408, 2022.

[Rahmani *et al.*, 2019] Hossein A Rahmani, Mohammad Aliannejadi, Rasoul Mirzaei Zadeh, Mitra Baratchi, Mohsen Afsharchi, and Fabio Crestani. Category-aware location embedding for point-of-interest recommendation. In *SIGIR*, pages 173–176, 2019.

[Rahmani *et al.*, 2022] Hossein A Rahmani, Mohammadmehdi Naghiaei, Ali Tourani, and Yashar Deldjoo. Exploring the impact of temporal bias in point-of-interest recommendation. In *RecSys*, pages 598–603, 2022.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, page 452–461, 2009.

[Saito and Joachims, 2021] Yuta Saito and Thorsten Joachims. Counterfactual learning and evaluation for recommender systems: Foundations, implementations, and recent advances. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pages 828–830, 2021.

[Singla *et al.*, 2023] Sumedha Singla, Nihal Murali, Forough Arabshahi, Sofia Triantafyllou, and Kayhan Batmanghelich. Augmentation by counterfactual explanation-fixing an overconfident classifier. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4720–4730, 2023.

[Tan *et al.*, 2021] Haining Tan, Di Yao, Tao Huang, Baoli Wang, Quanliang Jing, and Jingping Bi. Meta-learning enhanced neural ode for citywide next poi recommendation. In *MDM*, pages 89–98. IEEE, 2021.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pages 165–174, 2019.

[Wang *et al.*, 2020] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. Global context enhanced graph neural networks for session-based recommendation. In *SIGIR*, pages 169–178, 2020.

[Wang *et al.*, 2021a] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 373–381, 2021.

[Wang *et al.*, 2021b] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 347–356, 2021.

[Wu *et al.*, 2019] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, volume 33, pages 346–353, 2019.

[Yang *et al.*, 2013] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM conference on hypertext and social media*, pages 119–128, 2013.

[Yang *et al.*, 2021] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. Top-n recommendation with counterfactual user preference simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2342–2351, 2021.

[Zhang *et al.*, 2021] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 367–377, 2021.

[Zhao *et al.*, 2020] Guoshuai Zhao, Peiliang Lou, Xueming Qian, and Xingsong Hou. Personalized location recommendation by fusing sentimental and spatial context. *Knowledge-Based Systems*, 196:105849, 2020.