arXiv:2304.04151v1 [cs.IR] 9 Apr 2023

作为提示的时间戳：地理感知位置推荐

Yan Luo[1,2],Haoyi Duan[3], Ye Liu[1], Fu-lai Chung[1]1香港理工大学计算学系2麻省理工学院媒体实验室3浙江大学计算机科学与技术学院
{csyluo,cskchung}@comp.polyu.edu.hk, howie@zju.edu.cn,
coco.ye.liu@connect.polyu.hk

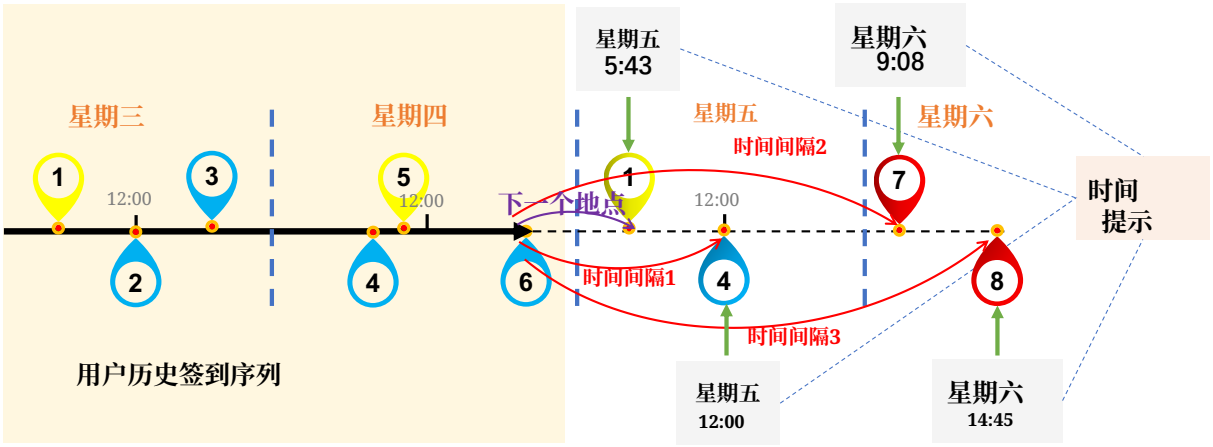


图1：ho 的示意图　　w TPG 执行下一位置推荐（以紫色线表示）并 inter
验证预测（以红线表示）　　）通过使用时间提示。不同颜色的标记表示不同类别的 P　　OIs.

摘要

位置推荐在改善用户的旅行体验方面发挥着至关重要的作用。待预测兴趣点的时间戳意义重大，因为用户会在不同时间前往不同地点。然而，大多数现有方法要么不使用这类时间信息，要么只是将其与其他上下文信息隐式融合。本文重新审视了位置推荐问题，并指出，显式建模时间信息大有裨益，尤其是在模型不仅需要预测下一个地点，还需要预测更远地点时。此外，最先进方法未能有效利用地理信息，并且在通过网格化编码地理信息时会遇到硬边界问题。为此，我们提出了一种基于时间提示且地理感知（TPG）框架。首先，我们设计了时间提示，以融入任意未来签到的时间信息；随后，我们设计了移位窗口机制，通过增强地理数据来解决硬边界问题。通过在五个真实世界数据集上与现有方法进行广泛比较并开展消融研究，我们证明了所提方法在各种设置下的有效性和优越性。最重要的是，所提模型具有出色的区间预测能力。具体而言，即使屏蔽了最新的签到行为数据，模型仍能预测用户在特定时间想去的地点；或者，它可以预测给定时间戳对应的特定未来签到（而不仅仅是下一个签到）。

关键词

兴趣点、位置推荐、序列推荐

1 引言

位置推荐是大数据时代的必然产物。一方面，Foursquare 和 Gowalla 等基于位置的社交网络（LBSN）的普及，带来了海量的用户签到数据。另一方面，尽管一座典型城市拥有数千个兴趣点，大多数用户无论是在家乡还是在外地，都会去的兴趣点都非常有限 [16, 21]。利用用户签到数据，位置推荐能够从众多兴趣点中为人们推荐最感兴趣的地点。位置推荐不仅能满足用户的出行需求，还具有巨大的商业价值，可应用于精准广告等领域。

大量研究致力于提升位置推荐的质量。一些研究采用推荐系统中最常用的解决方案——矩阵分解 [13, 35]。然而，这些方法没有考虑固有的时空序列行为，因此性能并不理想。传统的基于统计的马尔可夫链模型 [2, 6]已被广泛用于解决这一序列推荐问题，但它们存在局限，例如只考虑上一次签到活动的影响。随着数据量的增加以及深度学习技术的发展，循环

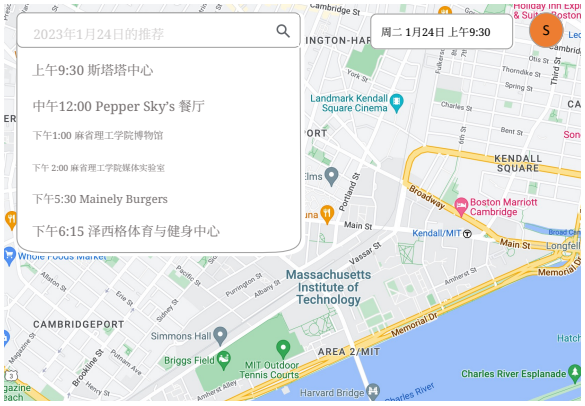


图2：TPG 在地图服务中的一种可能使用场景。左栏是TPG针对多个时间戳生成的推荐列表。右上方的橙色圆圈显示用户信息。橙色圆圈左侧的条形表示当前时间戳。

神经网络（RNN）方法 [4, 36] 被用于考量长期历史信息，其表示能力远强于基于马尔可夫链的模型。为解决基于矩阵分解的模型中的稀疏性问题，研究者提出了一些基于图的方法 [27, 29]，并取得了出色的效果。近期，研究者提出了注意力机制，能够有效建模人类轨迹中的长程时空依赖关系，从而用于位置推荐[12, 30]。对于位置推荐系统的输入，除了简单的签到序列，学者们还尝试使用多种信息来源，例如社交关系和地理信息[30]。

然而，这些方法中仍有两个主要挑战尚未得到解决。第一个是，大多数上下文感知的最先进方法都隐式地融入了与签到相关的时间信息。它们将签到的时间间隔信息或时间戳与签到历史相融合。当模型不仅需要预测下一个地点，还需要预测更后面的地点时，这会导致不灵活且不准确的兴趣点预测输出。例如，STAN [17] 为轨迹片段内的所有签到构建时空矩阵。CARA [18]同时利用签到序列以及与这些序列相关的上下文信息。如果我们希望基于前 100 个历史签到来预测第101 个签到和第102个签到，那么用于第 101个和第 102 个签到预测的输入（即，前100 个历史签到及其相关的上下文信息）是相同的。换句话说，这些模型没有考虑待预测地点的不同时间戳。然而，事实是，人们往往会在不同的时间前往不同的地点。例如，用户通常早上去上班，晚上回家。因此，对于位置推荐系统而言，具备基于特定时间戳生成特定预测的能力非常重要。值得注意的是，诸如 STAN和 CARA 之类的方法当然可以通过将第 101 个预测签到纳入前100条历史签到。然而，这样一来，这类模型都需要重新训练自身，这将需要大量计算开销。

第二个主要挑战是，地理信息对于位置推荐非常重要，因为这是一个时空问题。最先进方法未能有效利用地理信息。MobTCast [30] 将兴趣点的经纬度直接输入编码器。然而，由于签到数据极其稀疏， [1], 以这种方式处理地理信息很难捕捉地点间的物理邻近性和依赖关系。GeoSAN [12] 进一步提出使用层次网格来建模人类移动中的空间聚集现象。然而，它存在硬边界问题，也就是说，位于网格边界附近的兴趣点会被人为地分开。

为解决这些问题，我们提出了一种基于时间提示且地理感知（TPG）的位置推荐框架。我们将 TPG 设计为基于 Transformer 的框架，因为 Transformer [25]最初是为长度不确定的序列数据设计的，能够区分不同签到的信息量，并同时聚合轨迹中的所有签到以进行预测。首先，我们设计了一种地理感知编码器，用于捕获兴趣点之间的地理关联。为避免上述硬边界问题，我们在地理感知编码器中提出了一种移位窗口机制。该机制可以弥合两个相邻网格之间的邻近间隔，并通过聚合将相邻网格连接起来。随后，我们通过历史编码器整合用户历史签到序列中的用户、兴趣点、时间和地理信息，该编码器旨在学习用户出行偏好的全面表征。之后，我们将时间戳用作提示并视为查询，利用基于时间提示的解码器预测未来位置。通过这种方式，TPG 显式地纳入待预测位置的时间戳，将历史签到序列与时间戳信息分离。因此，TPG 能够灵活适用于多种场景。它不仅可以进行下一位置推荐，还可以利用基于未来位置的时间提示进行区间预测。需要指出的是，区间预测有两种等价场景：(a) 基于固定长度的历史轨迹片段（例如，前100次签到）预测之后的若干次签到（例如，第102次、第103次）；(b) 在屏蔽最近的签到行为数据时（例如，仅使用前95次或96次签到），预测一个未来位置（例如，第100次）。图1是一个简单示例，展示了 TPG 如何进行下一位置推荐和区间预测。如图1所示，给定用户从星期三到星期四的历史签到序列为兴趣点1至6，模型可以得知用户接下来将访问的四个地点分别是星期五5:43的兴趣点1、星期五12:00的兴趣点4、星期六9:08的兴趣点7，以及星期六14:45的兴趣点8。预测星期五5:43的兴趣点1，就是下一位置推荐任务。借助时间提示，TPG 还可以预测用户希望在特定时间前往的地点（即，区间预测）。例如，仅基于历史签到序列兴趣点1至6，模型就可以预测星期五12:00的兴趣点4（时间间隔1）、星期六9:08的兴趣点7（时间间隔2）和星期六14:45的兴趣点8（时间间隔3）。图2展示了现实世界应用的一种可能用例。

总结而言，本文的贡献可列举如下：

- 我们认为，在现实世界应用中，显式建模待预测地点的时间戳至关重要。本文提出了一种新颖且有效的基于 Transformer 的框架，名为 TPG。我们将时间信息视为推荐系统的提示。
- 为有效利用地理信息，我们提出了一种地理感知编码器，并设计了移位窗口机制，以避免在使用网格处理兴趣点的经度和纬度时出现硬边界问题。
- 在五个真实世界数据集（Gowalla、Brightkite、Foursquare-NYC、Foursquare-TKY 和 Foursquare-SIN）上的实验结果表明，在不同设置下，我们的模型均优于最先进的同类方法。我们还证明，TPG 的区间预测表现远胜于基线方法。

2 相关工作

本节首先回顾序列推荐和下一位置推荐方面的近期研究，然后进一步介绍自然语言处理（NLP）领域基于提示的学习的最新进展。

2.1 序列推荐

最初基于马尔可夫链 [2, 6], 的序列推荐模型能够通过转移矩阵预测下一行为发生的概率。由于马尔可夫模型存在缺陷，即主要考虑相邻两次访问之间的转移概率，因此基于深度学习的模型应运而生。其中，以循环神经网络 [34] 模型为代表的方法发展迅速，已成为强大的基线方法。同时，其他深度学习方法也展现出重要贡献，例如度量嵌入算法 [5], 、卷积神经网络 [23],、强化学习算法 [19], 和图神经网络 [28, 32] 。近年来，得益于完全并行化以及捕获长程依赖关系的能力，自注意力网络 [25]被用于序列推荐，并取得了优异的表现。例如，GeoSAN [12] 采用自注意力层，达到了最先进的性能。

然而，上述大多数方法仅通过融合历史签到序列及其相关上下文信息来预测下一个位置，忽略了待预测位置的时间戳也具有重要意义。所提出的 TPG 框架通过一种新颖的时间提示对时间信息进行显式建模。

2.2 下一位置推荐

下一位置推荐，即，兴趣点推荐，近年来因其巨大的商业价值而受到相当多的注意力机制，可以被视为带有空间信息的序列推荐的一个特殊子任务 [17]。关于在下一位置推荐中使用时空信息，许多先前的工作只在循环层中使用两次连续访问之间的时空间隔。例如，DeepMove [4] 结合了一个注意力层，用于学习长期的序列规律。LSTPM [22] 提出了一种地理扩张RNN，用于聚合近期访问的地点，但仅针对短期偏好。受序列物品推荐 [9], 启发，GeoSAN [12] 在轨迹内使用自注意力模型进行下一个地点推荐。STAN[17] 采用了一种时空注意力网络，用于聚合轨迹中的所有相关签到。GETNext [31] 通过利用广泛的协同信号，提出了一种新颖的图增强 Transformer 模型。

然而，这些模型在地理和时间信息建模方面存在局限。在 TPG 中，我们应对网格映射中的硬边界问题，并提出了移位窗口机制。我们还摒弃了 TPG 中融合时间信息的隐式方式。我们提出了时间提示，用于显式建模待预测地点的时间戳。

2.3 基于提示的学习

近年来，我们见证了预训练模型的成功，例如 GPT系列，尤其是 GPT3 [3], ，这促使基于提示的学习范式，即“预训练、提示和预测” [14], 在自然语言处理任务上蓬勃发展。此时，模型不再被设计为适应任务。相反，各种任务通过模板进行修改，以便更接近预训练模型的空间，而预训练模型则使用来自网络的大量语言数据进行训练。在整个发展过程中，许多提示设计方法不断涌现。其中一种方法 [7] 是设计合适的离散提示，专注于模板的设计。同时，另一类工作 [15] 利用连续向量嵌入作为提示，以提升自然语言处理任务上的性能。此外，作为另一种类型，基于指令的提示包含详细的任务描述，并且更遵循自然语言格式。

事实上，推荐早已与自然语言处理技术共同发展了很长时间。受这些方法在自然语言处理任务上取得成功的启发，研究人员近期将基于提示的学习的核心思想迁移到推荐问题中，并取得了非常有前景的结果。P5 [8] 构建了一组个性化提示，然后根据所构建的个性化提示，将各种与推荐相关的任务转换为自然语言表达，并在这些任务上训练一个序列到序列模型。

据我们所知，我们的工作首次在基于自注意力机制的序列推荐框架中，将时间戳用作位置推荐的提示，从而显式地建模时间信息。

3 新框架

本节将详细介绍所提出的 TPG 框架。我们首先给出问题陈述，并概述该框架。然后，详细阐述 TPG 的三个主要模块，即 地理感知编码器、历史编码器和基于时间提示的解码器。

3.1 概述

每条签到 $c_i^u = (u, t_i, p_i)$ 都是一个用户、兴趣点、时间三元组，表示用户 u在时间 t_i p_i访问了兴趣点。每个兴趣点 p_i都有自己的地理坐标（x_i, y_i）。每个用户 u都有一串历史签到 $C_{1\to n}^u = \{c_i^u\}_{i=1}^n$。给定历史

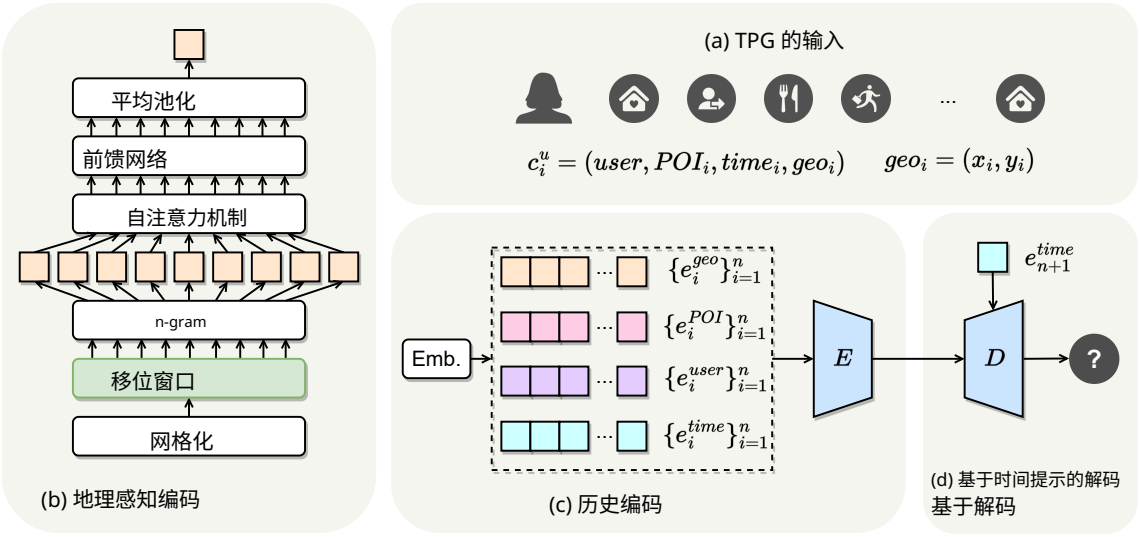


图 3：所提出的 TPG 的整体架构。符号的详细说明见第 3 节。

用户的签到序列，下一个地点推荐的目标是预测下一个兴趣点 $\rho_{t_{n+1}}$，某个用户 u 将在某个时间 t_{n+1}到访。

我们的 TPG 框架整体架构如图 3 所示。基于 Transformer 的编码器—解码器结构，TPG 可分为三部分，即 地理感知编码器、历史编码器和基于时间提示的解码器。对于每次签到，兴趣点的地理坐标可输入地理感知编码器，以获取地理表示 e_i^{geo}。随后，将包含兴趣点、用户和时间信息的历史签到序列输入多模态嵌入模块，以生成隐藏表示 $\{e_i^{POI}\}_{i=1}^n$、$\{e_i^{user}\}_{i=1}^n$ 和 $\{e_i^{time}\}_{i=1}^n$。这些表示与地理感知编码器生成的 $\{e_i^{geo}\}_{i=1}^n$ 共同输入历史编码器，以生成用户出行偏好表示。以 t_{n+1} 的时间信息作为提示后，时间提示查询和用户出行偏好记忆将被传递至解码器，从而更准确地预测下一个地点。

3.2 地理感知编码器

稀疏性问题是推荐问题中的一项关键挑战。特别是，由于签到数据提供了访问行为的隐式反馈 [13], 对于兴趣点而言，数据稀疏性问题比电影或商品等其他项目推荐更加严重，因为用户通常只会通过评分表达自己的意见。因此，在对兴趣点的地理信息进行编码时，直接将兴趣点的坐标输入学习模型，会使模型难以捕捉地理关联。GeoSAN[12] 通过使用瓦片地图系统[1]将纬度和经度映射到层次网格中，从而嵌入地点的精确位置，如图 4所示。瓦片地图系统是一种分层的多分辨率金字塔模型。世界地图通过投影获得通过墨卡托[2]将整个世界转换为平面。该平面的比例尺从 512×512 像素开始。随着层级增加，比例尺以2为因子增长。为了更好地进行检索和显示，平面进一步划分为每个大小为 256×256 像素的网格。从瓦片金字塔的低层级bottom到高层级top，通过将一个网格细分为四个相同大小的网格，分辨率变得越来越低，但地理范围保持不变。由于网格的划分方式类似于四叉树，因此每个网格都可以通过唯一的四叉树键进行标识（简称四叉树键）。四叉树键由集合{“0”、“1”、“2”、“3”}中的字符组成。其长度等于网格的层级。

这确实能在一定程度上缓解稀疏性问题。然而，对于同一层级的网格，网格边界可能会破坏边界附近两个兴趣点在物理空间上的邻近性，这违背了托布勒的地理学第一定律 [24]。换言之，缺少跨相邻网格的连接会限制地理感知编码器的建模能力。为了引入跨网格连接，我们在地理感知编码器中提出了一种移位窗口机制。如图 5 所示，对于每个网格，我们沿 X 轴和 Y 轴方向（以及两个方向同时）将移位窗口移动一定步长，该步长是网格尺寸的一部分。这样，每个网格都会得到九个网格，即 其自身和八个增强邻居网格。

现在，地理感知编码器剩下的任务是将这九个网格的四叉树键转换为包含丰富信息的连续潜在嵌入。每个四叉树键实际上都是一个字符序列。序列中的每个字符表示某一层级上网格划分的索引。若要有效利用网格的层次化空间信息，一个直观直接的方法是在这些字符之间进行自注意力计算。然而，由于字符集的基数非常小（即，只有 4 个），将四叉树键按字符级别处理

[1]https://www.maptiler.com/google-maps-coordinates-tile-bounds-projection

[2]https://www.britannica.com/science/Mercator-projection



图4：基于瓦片地图系统的分层网格化示意图。给出的示例展示了如何将一个位置映射到第16至18级网格中，并标注了相应的四叉树键。

无法实现充分编码兴趣点之间地理关联的目标。因此，我们考虑按 n-gram 划分字符序列，并将其转换为 n-gram 级别的序列。这样一来，序列的词汇表大小便从 4 增至 4^n。例如，若四叉树键为“013201233”，使用四元组得到的结果为 0132-1320-3201-2012-0123-1233。随后，我们使用堆叠自注意力网络和逐点前馈网络来捕捉这些 n-gram 之间的依赖关系。之后，对于这九个网格中的每一个，都可以通过平均池化汇总 n-gram 表示序列。然后，对这九个网格的汇总表示进行平均池化，即可获得给定位置POI_i的地理嵌入 e_i^{geo}。移位窗口机制通过纳入增强邻居网格的信息，有助于减少网格任意划分可能造成的偏差。

3.3 历史编码器

每次签到都是由用户、时间和兴趣点信息组成的元组。为了处理这类离散且异构的数据，我们需要一个多模态嵌入模块，将签到数据转换为可供TPG 解读的信息。具体而言，在编码时间信息时，首先将时间戳离散化为 $24 \times 7 = 168$ 类可学习向量。在编码用户和兴趣点信息时，可学习向量的类型数量分别等于数据集中唯一用户和兴趣点的数量。随后，所有这些向量都被线性投影为 d维嵌入 $e_i^{time} \in \mathbb{R}^d$、$e_i^{user} \in \mathbb{R}^d$ $e_i^{POI} \in \mathbb{R}^d$ u 和 。这样，对于用户，历史签到序列 $\{c_i^u\}_{i=1}^n$可进一步表示为 $(\{e_i^{POI}\}_{i=1}^n, \{e_i^{user}\}_{i=1}^n, \{e_i^{time}\}_{i=1}^n, \{e_i^{geo}\}_{i=1}^n)$。请注意，由于签到数据需要遵循一定的先后顺序，因此还会将可学习的位置嵌入添加到历史编码器的输入中。

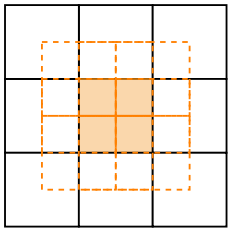


图 5：橙色标记的网格移位窗口示意图。此处，我们将滚动步长设为网格边长的一半。

与以往基于RNN的方法相比，Transformer架构 [25]不仅能够避免循环运算，实现并行计算以缩短训练时间，还能缓解RNN中长期依赖方面的性能下降问题。为了更好地捕捉用户历史签到序列中的长程时空依赖，我们堆叠Transformer编码器层 [25] 来构建历史编码器。每个Transformer编码器层都包含一个多头自注意力模块和一个逐点前馈网络。我们还保留了Transformer编码器层中使用的残差连接和层归一化。将注意力机制划分为多个头以形成多个子空间，可以使模型关注信息的不同方面。对于每个注意力头，签到 c_i 的自注意力结果可计算为

$$ATTENTION(e_i^c) = w_z \sum_{j=1}^{N_v} \frac{\exp(w_q e_i^c \times w_k e_j^c)}{\sum_{m=1}^{N_v} \exp(w_q e_i^c \times w_k e_m^c)} w_v e_j^c + e_i^c \tag{1}$$

其中，e_i^c 是 c_i 的输入签到嵌入，e_j^c 是序列中上下文签到的嵌入，而 $w_{\{q,k,v,z\}}$ 表示查询、键、值和输出矩阵的线性变换权重。自注意力机制将全局上下文信息聚合到每个签到特征中。通过拼接所有自注意力结果得到多头自注意力结果后，编码器能够联合关注不同位置、不同表示子空间中的信息。随后，前馈网络包含两个线性变换，中间采用 ReLU 激活函数，可表示为

$$FFN(e_{i'}^c) = max(0, e_{i'}^c W_1 + b_1) W_2 + b_2 \tag{2}$$

其中，$e_{i'}^c$ 是经过多头自注意力处理后的输入嵌入，W_1和 W_2 表示线性变换权重，b_1 和 b_2 是线性变换偏置。

3.4 基于时间提示的解码器

现有方法通常根据历史签到及其关联的上下文信息来生成下一个地点的预测结果。模型不会明确针对特定时间信息给出输出。然而，在实际应用中，进行下一个地点预测时，考虑确切时间非常重要。位置推荐通常用于 Google 地图等地图服务，以及 Foursquare 等基于位置的服务。我们可以将用户打开

应用或点击应用中查询框的时间视为下一个地点的时间戳。人类移动具有周期性 [33]。人们会在不同日期的同一时段再次访问同一类别的兴趣点。而在一天中的不同时段，人们往往会前往不同类型的兴趣点。例如，用户在中午点击地图应用中的查询框，很可能是在寻找餐厅，这是显而易见的事实。而在早上，他/她很可能想搜索前往其公司的路线。因此，如果模型不知道下一个地点的时间戳，就无法高置信度地给出下一个地点的预测结果。由此，一个直观的想法是直接告知模型下一个地点的时间戳。一种简单的方法是将此时间戳纳入输入。然而，由于签到是由用户、时间和兴趣点组成的元组，直接将这一时间戳添加到输入中并不合适。为此，我们提出了一种基于时间提示的解码器，将时间戳用作解码器的提示和查询。使用时间提示有以下几项优势：(1) 它将历史签到序列与待预测地点的时间信息分离，使模型能够更灵活地生成任意未来签到。(2) 通过显式建模时间信息，预测结果与给定时间戳的关联性大大增强。如上所述，人们的出行选择与时间戳密切相关，因此这一设计很可能提升模型性能。

更具体地说，偏好解码器将查询（即 时间表示 e_{n+1}^{time}）和编码器记忆（即 用户旅行偏好表示 e^{C}）作为输入。我们通过堆叠 Transformer 解码器层[25],来构建基于时间提示的解码器，每层均由一个多头自注意力子层、一个编码器-解码器注意力子层和一个前馈网络子层组成。在每个编码器-解码器注意力子层中，用户旅行偏好和时间戳信息得到了深度融合。编码器-解码器注意力子层中的每个注意力头可表示为

$$ATT(e^{C}, e_{n+1}^{time}) = w_z \sum_{j=1}^{N_v} \frac{\exp(w_q e^{C} \times w_k e_{n+1}^{time})}{\sum_{m=1}^{N_v} \exp(w_q e^{C} \times w_k e_{n+1,m}^{time})} w_v e_{n+1}^{time} + e^{C} \tag{3}$$

其中，符号与公式 1 中的定义一致。

将结果相加并输入前馈网络进行进一步投影后，输出嵌入会解码融合后的签到特征，其长度与查询嵌入相同。此处的输出实际上是预测的下一个地点的嵌入。

在训练方案方面，我们采用带采样 Softmax 的负对数似然作为每个用户 u的推荐损失。推荐损失可表示为：

$$\mathcal{L}_{rec}(\tilde{y}) = -\log \frac{\exp(\tilde{y}y^{+})}{\exp(\tilde{y}y^{+}) + \sum_{y^{-} \in \mathcal{Y}^{-}} \exp(\tilde{y}y^{-})} \tag{4}$$

其中， $\tilde{y}$、 y^{+} 和 y^{-} 分别表示推断得到的位置嵌入、用户 u 在时间 t_{n+1} 访问的位置真实值，以及用户在 t_{n+1}未访问的位置随机采样的负样本。

表 1：数据集统计信息。

	Gowalla	Brightkite	NYC	TKY	SIN
#用户	31,708	5,247	1,010	6,771	367
#地点	131,329	48,181	5,135	14,590	3,104
#签到	2,963,373	1,699,579	140,229	871,200	136,847

4 实验与评估

本节报告了为评估性能而开展的大量实验，并展示了我们方法的优越性能。实验结果还证明了 TPG 的有效性和实用性。进一步的实验也验证了 TPG 各个组成部分的合理性。

4.1 实验设置

4.1.1 数据集。 我们使用五个公开的真实世界基于位置的社交网络数据集来评估我们的方法：Gowalla3、Brightkite4、纽约市、东京和新加坡[6]。Gowalla 和 Brightkite 包含全球范围的数据，而纽约市、东京和新加坡数据则提取自 Foursquare 全球数据集，该数据集仅关注单个城市或地区。我们采用了 LibCity 的 [26] 预处理流程。表 1 概述了五个数据集的统计信息。

4.1.2 评估指标。 我们采用两种广泛使用的排序评估指标：召回率和 NDCG，来评估推荐性能。Recall@k 统计所有正样本中真正样本的比例，在我们的场景中，即标签出现在前 k 个概率样本中的比例；NDCG 则奖励将正样本排在前 k 排名列表前几位的方法。我们在实验中报告 k=5 和 k=10 。

4.1.3 基线方法。为展示我们提出的方法的有效性，我们将所提出的 TPG 与以下几种基线方法进行比较：

- HSTLSTM [10]: 一种基于 LSTM 的方法，引入了时空迁移因子，并采用编码器-解码器结构进行预测。
- DeepMove [4]: 一种注意力循环网络，能够捕捉复杂的序列转移和多层级周期性。
- LSTPM [22]: 一种长短期偏好建模框架，由用于长时偏好建模的非局部网络和用于学习短期偏好的地理扩张 RNN组成。
- CARA [18]: 一种新颖的上下文注意力循环架构，利用反馈序列及其相关上下文信息来捕捉用户的动态偏好。
- TMCA [11]: 一种新颖的基于时间和多层次上下文注意力 LSTM 的编码器-解码器框架，能够

[3]https://snap.stanford.edu/data/loc-gowalla.html [4] http://snap.stanford.edu/data/loc-brightkite.html [5] https://drive.google.com/file/d/0BwrgZ-IdrTotZ0U0ZER2ejI3VVk/view?usp=sharing&resourcekey=0-rlHp_JcRyFAxN7v5OAGldw [6] https://www.ntu.edu.sg/home/gaocong/data/poidata.zip

表2：与八种基线方法的比较。最佳结果以粗体标出，次佳结果以下划线标出。

	Gowalla				Brightkite				NYC				TKY				SIN			
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
HSTLSTM	44.6%	30.9%	54.4%	35.1%	46.8%	37.3%	50.4%	41.3%	25.7%	19.7%	31.1%	21.4%	29.1%	22.1%	35.7%	24.3%	14.9%	11.2%	21.0%	13.4%
DeepMove	49.9%	35.9%	59.1%	39.0%	50.3%	39.1%	58.7%	43.2%	29.8%	21.6%	36.5%	23.8%	33.3%	25.1%	39.8%	27.2%	16.8%	11.8%	24.7%	13.4%
LSTPM	42.2%	30.4%	53.2%	33.0%	43.8%	34.9%	52.4%	38.5%	22.8%	16.1%	30.7%	18.6%	37.2%	28.2%	45.4%	30.9%	13.1%	9.2%	20.9%	12.0%
TMCA	44.3%	32.5%	55.4%	35.5%	45.5%	36.4%	55.9%	41.6%	24.6%	18.3%	33.1%	20.2%	39.9%	27.9%	41.7%	29.5%	15.5%	11.3%	22.3%	14.0%
CARA	50.2%	36.6%	60.0%	40.8%	51.6%	40.2%	54.7%	41.2%	28.0%	20.2%	37.5%	24.0%	31.8%	24.3%	37.2%	28.0%	15.04%	11.8%	20.8%	14.0%
MobTCast	54.3%	37.9%	65.5%	46.6%	52.5%	43.6%	59.5%	46.4%	31.3%	21.3%	41.3%	28.1%	59.7%	48.4%	65.4%	52.6%	17.0%	12.4%	24.6%	15.1%
STAN	58.7%	41.6%	70.3%	46.6%	57.2%	45.4%	69.8%	47.3%	32.1%	22.3%	45.9%	27.3%	61.2%	50.1%	71.3%	54.4%	18.0%	11.3%	29.5%	15.7%
GeoSAN	56.2%	41.4%	69.9%	45.9%	55.8%	42.3%	67.2%	46.0%	30.2%	20.8%	44.5%	25.3%	58.8%	48.4%	69.1%	51.8%	17.4%	11.8%	30.0%	15.9%。
TPG	**63.2%**	**45.5%**	**74.6%**	**50.1%**	**62.4%**	**47.9%**	**74.4%**	**51.8%**	**37.3%**	**26.8%**	**52.8%**	**31.8%**	**65.4%**	**53.1%**	**76.5%**	**56.7%**	**19.4%**	**14.3%**	**34.3%**	**19.1%**
改进	7.7%	9.4%	6.1%	7.5%	9.1%	5.5%	6.6%	9.5%	16.2%	20.2%	15.0%	13.2%	6.9%	6.0%	7.3%	4.2%	7.8%	15.3%	14.3%	20.1%

自适应地选择与下一兴趣点偏好预测相关的签到活动和上下文因素

- GeoSAN [12]：一种具有地理感知能力的序列推荐器，基于自注意力网络，使用 GPS 地点的分层网格化进行空间离散化，并使用自注意力层。
- STAN [17]：一种时空注意力网络，明确聚合轨迹中所有相关签到，而不仅仅是连续的签到。
- MobTCast [30]：一种基于 Transformer 的上下文感知网络，结合了作为辅助任务的位置预测分支。该网络能够捕捉时间、语义、社交和地理上下文。

4.1.4 实现细节。 对于每位用户的签到序列，我们在评估时将其在此前未访问过的地点上的最后一条签到记录作为真实值，并使用此前的签到序列进行训练。序列最大长度设为 100。

与 GeoSAN 在训练和评估阶段都直接使用真实值进行负采样、可能导致标签泄漏不同，我们考虑一种实际场景：每位用户的下一个实际位置未知，因此负样本必须从紧邻的上一签到地点附近采集。具体而言，从用户当前 GPS 坐标最近的 2000 个地点中随机选取 100 个作为负样本。随后，可根据这 101 个地点的排序计算召回率和 NDCG。

我们所有实验均在 NVIDIA V100 GPU 上运行。对于 TPG 模型，我们分别将位置和区域嵌入的维度设为 50，将时间嵌入的维度设为 100。移位窗口的步长设为网格边长的四分之一。我们使用学习率为 0.001 的 Adam 优化器训练模型，并将 dropout 比率设为 0.5。四个数据集的训练轮数均设为 50。对于除 STAN 以外的基线方法，我们遵循其实现方式以及论文中声称的最佳设置。STAN 会为每个用户的所有历史签到构建矩阵，这会消耗大量时间和内存。在配备 768GB 内存的服务器上运行原版 STAN 时，出现了内存溢出（OOM）错误。为了测试 STAN 的性能，我们选择部分用户进行训练，并每次针对这些用户训练模型并测试其性能。对于纽约市和新加坡数据集，我们使用所有用户。对于其他数据集，由于用户数量较多，我们选取前 2000 名用户来测试模型性能。

4.2 整体性能比较

我们将所提出的 TPG 与上述基线方法进行性能比较。表 2 报告了 TPG 和八种基线方法在五个真实世界数据集上的 Recall@k 和 NDCG@k 性能。“Improv.”列表示 TPG 相较于第二佳模型的改进率。根据结果，我们观察到：

我们的 TPG 在所有数据集上，无论是 NDCG@k 还是召回率@k，以及 k 的所有取值方面，都显著优于所有基线方法。与表现最佳的基线方法相比，我们提出的方法在 NDCG@5 和召回率@5 方面的提升幅度最高分别达到 20.2% 和 16.2%。这证明了我们提出的 TPG 的有效性和优越性；该方法有效利用了下一个地点的地理信息和时间信号。

(2) 与基于RNN的方法相比，纯注意力机制方法（如 MobTCast、STAN、GeoSAN 以及我们提出的 TPG）显然表现更佳。这是合理的，因为注意力机制能够捕获时空签到序列中的全局上下文信息，而基于RNN的方法则面临遗忘过去远距离信息的风险。在基于RNN的模型中，DeepMove 和 CARA 的表现通常优于其他模型，这归功于它们对时空建模以及短期和长期周期性建模的考量。这些设计在一定程度上弥补了 RNN固有的缺陷。与先进的注意力机制模型 MobTCast、STAN 和 GeoSAN 相比，TPG取得的显著提升证明了明确利用下一个地点的时间信号以及用于地理网格化的移位窗口机制的重要性。尽管 STAN 的表现通常优于 MobTCast 和 GeoSAN，但由于它需要对所有历史签到执行矩阵运算，因此会产生极大的内存和计算时间开销。无论在计算时间、内存还是准确率方面，我们的 TPG 方法都远胜于其他方法。

表 3：区间预测性能。我们将高于 TPG 的区间预测结果加粗。

	Gowalla				Brightkite				NYC				TKY				SIN			
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
STAN	58.7%	41.6%	70.3%	46.6%	57.2%	45.4%	69.8%	47.3%	32.1%	22.3%	45.9%	27.3%	61.2%	50.1%	71.3%	54.4%	18.0%	11.3%	29.5%	15.7%
int. 1	54.2%	36.5%	66.5%	42.2%	52.1%	40.6%	66.2%	43.3%	30.0%	20.3%	42.5%	24.5%	57.4%	46.9%	67.7%	52.1%	16.8%	10.3%	27.5%	14.4%
int. 2	48.2%	31.3%	62.2%	37.0%	50.4%	40.9%	65.2%	42.0%	24.6%	14.5%	36.5%	18.3%	52.5%	43.8%	64.4%	49.4%	13.5%	8.3%	25.5%	13.4%
int. 3	42.4%	28.1%	58.4%	33.3%	47.0%	37.5%	63.3%	40.3%	20.3%	10.9%	32.1%	15.6%	47.9%	40.2%	62.5%	45.9%	12.1%	8.5%	23.6%	12.1%
GeoSAN	56.2%	41.4%	69.9%	45.9%	55.8%	42.3%	67.2%	46.0%	30.2%	20.8%	44.5%	25.3%	58.8%	48.4%	69.1%	51.8%	17.4%	11.8%	30.0%	15.9%
int. 1	54.3%	39.5%	67.8%	43.9%	52.1%	40.6%	66.2%	43.3%	29.0%	18.6%	43.4%	23.2%	56.3%	46.3%	67.7%	49.3%	16.4%	**12.0%**	29.5%	14.4%
int. 2	53.2%	39.6%	65.3%	41.2%	50.4%	40.9%	65.2%	42.0%	28.7%	18.3%	42.6%	22.9%	54.3%	44.8%	65.4%	47.6%	15.2%	10.4%	28.6%	13.4%
int. 3	50.1%	35.4%	62.1%	38.5%	47.0%	37.5%	63.3%	40.3%	25.7%	16.9%	39.6%	21.0%	55.3%	43.2%	65.2%	46.2%	14.2%	9.4%	26.4%	13.8%
TPG	63.2%	45.5%	74.6%	50.1%	62.4%	47.9%	74.4%	51.8%	37.3%	26.8%	52.8%	31.8%	65.4%	53.1%	76.5%	56.7%	19.4%	14.3%	34.3%	19.1%
int. 1	**63.7%**	44.3%	73.7%	49.1%	60.4%	46.2%	73.0%	50.3%	**38.0%**	26.7%	**53.6%**	**32.1%**	64.5%	52.9%	75.6%	56.4%	**19.9%**	13.3%	33.1%	18.1%
强度 2	60.5%	**45.9%**	72.8%	48.6%	59.9%	45.6%	72.5%	49.9%	**37.5%**	26.5%	**53.2%**	30.1%	64.1%	52.2%	75.1%	55.8%	19.1%	12.6%	32.2%	18.7%
int. 3	59.7%	**46.5%**	72.0%	47.1%	59.8%	45.3%	72.3%	49.8%	37.0%	25.3%	52.0%	29.7%	64.9%	**53.1%**	75.1%	**56.7%**	**19.6%**	**14.3%**	31.3%	17.6%

(3) 不同数据集的签到记录密度可表示为 #签到数 / (#用户 ×# 地点)。Gowalla 、 Brightkite 、纽约市、东京和新加坡的稀疏度分别为 0.001、0.007、0.027、0.009 和 0.120。这可以解释为什么 TPG 在纽约市和新加坡带来的提升大于在东京的提升。此外，显然 TPG 擅长处理 Gowalla 这类稀疏数据。

4.3 区间预测性能

通过引入时间提示，TPG 能够预测地点的准确时间戳，从而进行区间预测。相关结果见表 3。我们在所有数据集上，分别屏蔽用户最近的一次（表 3 中的“int. 1”）、两次（表 3 中的“int. 2”）和三次（表 3 中的“int. 3”）签到，以测试 TPG 以及两个基线方法 STAN 和 GeoSAN 的表现。具体设置为：使用每位用户轨迹中的前 96、97、98 次签到，预测第 100 次签到。

对于 STAN 和 GeoSAN，我们观察到，与使用全部签到数据相比，被掩盖的最新签到记录越多，性能下降得越明显，但幅度较小。这是合理的，因为一些先前的研究 [20] 已经证明，用户最近的行为对用户的下一个行为有很大影响。然而，对于 TPG 的区间预测，情况则完全不同。其性能有时甚至优于下一个地点推-荐。这些令人印象深刻的结果有力地证明了使用时间戳作为提示来显式建模时间信息的益处。

4.4 消融研究

我们对 TPG 进行了广泛的消融实验，以深入探究所提出模型中各模块的有效性。需要强调的是，我们的基础模型（表 4 第一行中记作 TPG）不包含用户嵌入。我们针对模型考虑了以下变体进行消融：

- 移除 TP（基于时间的提示）：我们使用地点作为解码器的查询，而不是使用基于时间的提示
- 移除TE（时间嵌入）：我们移除签到序列的时间嵌入，仅使用兴趣点嵌入和地理嵌入作为历史编码器的输入。
- 移除 SW（移位窗口机制）：我们移除了地理感知编码器中的移位窗口机制。
- RemoveGE（地理编码器）：我们移除地理编码器，仅使用兴趣点嵌入和时间嵌入作为历史编码器的输入。
- 添加 UE（用户嵌入）：我们将用户嵌入与兴趣点嵌入、地理嵌入和时间嵌入进行拼接，并将其添加到历史编码器中。

性能对比如表 4 所示。通过对比，我们得出以下几点发现：

（1）在没有基于时间的提示时，模型的整体性能会下降，尤其是在具有局部区域的 Foursquare 数据集上。这一现象表明，显式引入下一个地点的时间信号具有重要意义。此外，在 Gowalla 和 Foursquare 数据集上，“Remove TE” 的结果通常略好于“Remove TP”，但仍然逊于 TPG。这表明，如果不能恰当地使用时间信息，就会给模型带来额外噪声并降低其性能。同时，这也证明了本文提出的基于时间的提示策略能够有效利用时间信息。

（2）在大多数情况下，“Remove SW”的结果都差于原始 TPG。这表明，我们提出的移位窗口机制是空间信息的一种有效增强方法。它能够弥合相邻网格在空间分布方面的语义差距，并提高模型的准确性和稳定性。“Remove GE”的性能甚至差于“Remove SW”。这是合理的，因为地理信息对于基于位置的应用具有重要意义。

(3)“添加UE”通常会导致与 TPG 相比性能大幅下降。这是因为将用户嵌入添加到输入中

表 4：消融研究的性能。

	Gowalla		Brightkite		NYC		TKY		SIN	
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
TPG	63.2%	45.5%	74.6%	50.1%	62.4%	47.9%	74.4%	51.8%	37.3%	26.8%	52.8%	31.8%	65.4%	53.1%	76.5%	56.7%	19.4%	14.3%	34.3%	19.1%
I. – TP	60.4%	44.0%	73.0%	48.4%	61.1%	47.2%	73.6%	50.4%	34.3%	23.7%	47.1%	28.2%	62.2%	50.5%	72.9%	54.0%	18.3%	12.2%	29.7%	15.9%
II. – TE	61.7%	44.0%	73.7%	49.0%	57.3%	43.7%	69.0%	47.5%	36.9%	25.0%	50.8%	29.4%	65.4%	53.3%	76.4%	56.9%	21.3%	15.9%	32.2%	18.4%
III. – SW	58.77%	43.6%	72.6%	48.0%	61.5%	47.4%	73.7%	51.3%	36.8%	25.6%	52.3%	31.7%	64.0%	51.3%	75.1%	55.0%	19.9%	13.3%	30.5%	16.7%
IV. –GE	60.3%	42.6%	72.1%	47.6%	54.7%	42.2%	65.1%	45.6%	34.5%	25.2%	50.3%	28.50%	56.2%	43.3%	68.1%	47.4%	19.7%	14.3%	33.7%	17.1%
V. + UE	62.1%	45.1%	74.0%	48.5%	57.9%	44.1%	70.1%	48.0%	36.3%	25.4%	51.1%	32.0%	53.2%	40.1%	65.9%	44.3%	18.5%	13.1%	26.7%	15.7%

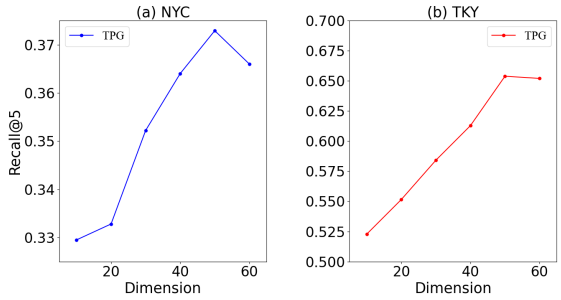


图 6：地理嵌入维度对模型性能的影响。

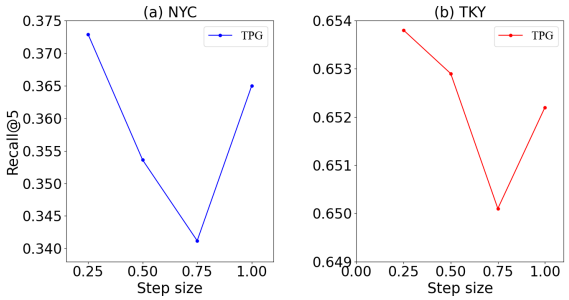


图7：移位窗口步长对模型性能的影响。

历史编码器可能会导致签到序列的向量空间与地点的向量空间之间出现错位。这表明，未来的工作可以适当地利用用户信息来提升性能。

4.5 参数敏感性分析

我们对两个重要参数进行敏感性分析，即地理嵌入维度和滑动窗口的步长。首先，我们研究模型对地理嵌入维度的敏感性。我们将地理感知编码器中使用的维度从 10 调整到 60，步长为 10。图 6 展示了在两个数据集纽约市和东京上的实验结果。由此可以得出结论：较小的地理嵌入维度会导致模型性能很差。这是因为较小的维度难以描述兴趣点之间复杂的地理关系，从而造成严重的信息损失。当维度为 50 时，模型性能达到峰值。维度增加到 60 时，性能略有下降。这可能是因为地理感知编码器构建的语义空间大小是有限的。当嵌入维度过高时，信息无法充分利用，反而可能引入噪声。

我们进一步研究了模型对移位窗口步长的敏感性。我们将移位窗口机制中使用的步长从0.25调整到1，步长为0.25。需要注意的是，这里的步长是指移动长度与网格大小的比例。我们仍以两个数据集纽约市和东京为例。实验结果如图7所示。我们可以发现，

对于移位窗口机制，性能在较小的步长0.25时达到峰值，随后一直下降，直到步长为 0.75。这种现象符合地理学第一定律，即“一切事物都与其他事物相关，但邻近事物之间的关联比远距离事物之间的关联更紧密”。我们还可以观察到，当步长为 1 时，性能有所提升。当步长为 1时，模型实际上退化为在每个层级直接聚合网格自身的邻近网格。这一现象证明了以往方法的缺陷，即它们没有充分考虑相邻网格之间的相关性。

5 结论

在本文中，我们重新审视了位置推荐问题。我们发现，大多数方法要么忽略了这样一个前提：在现实世界的应用中，需要知道预测兴趣点的确切时间；要么将时间信息与历-史签到隐式融合。我们提出了TPG，一种基于时间提示且具备地理感知的框架，用于下一位置推荐。我们展示了如何将时间戳作为提示，显式建模待预测地点的时间信息。通过提出移位窗口机制，我们还展示了如何避免与签到地理坐标相关的硬边界问题。在五个基准数据集上的实验结果表明，与其他最先进方法相比，TPG具有优越性。结果表明，地点的时间信号具有重要意义。我们还通过消融研究证明，我们提出的移位窗口机制能够克服

以往方法在地理信息建模方面的缺陷。

关于未来工作，我们计划设计更加智能的提示。来自自然语言处理社区的大规模预训练模型已经展示出提示学习的无限潜力。我们可以考虑通过提示，将位置推荐与语言预训练模型相结合。

参考文献

[1] Gediminas Adomavicius 和 Alexander Tuzhilin。2005。迈向下一代推荐系统：现状综述与可能的扩展。*IEEE 知识与数据工程汇刊*（2005）。[2] Paul Baumann、Wilhelm Kleiminger 和 Silvia Santini。2013。时间与空间特征对下一地点预测算法性能的影响。载于*ACM*普适与泛在计算国际联合会议论文集（*UbiComp*）。449–458。[3] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。2020。语言模型是少样本学习者。载于神经信息处理系统进展（*NeurIPS*）。1877–1901。[4] Jie Feng、Yong Li、Chao Zhang、Funing Sun、Fanchao Meng、Ang Guo 和 Depeng Jin。2018。DeepMove：利用注意力循环网络预测人类移动。载于万维网会议论文集（*WWW*）。1459–1468。[5] Shanshan Feng、Lucas Vinh Tran、Gao Cong、Lisi Chen、Jing Li 和 Fan Li。2020。HME：一种用于下一兴趣点推荐的双曲度量嵌入方法。载于*ACM*国际信息检索研究与发展会议论文集（*SIGIR*）。1429–1438。[6] Sébastien Gambs、Marc-Olivier Killijian 和 Miguel Núñez del Prado Cortez。2012。使用移动马尔可夫链预测下一个地点。载于测量、隐私与移动性研讨会论文集（*MPM*）。1–6。[7] Tianyu Gao、Adam Fisch 和 Danqi Chen。2021。让预训练语言模型成为更好的少样本学习者。载于计算语言学协会年会论文集（*ACL*）。3816–3830。[8] Shijie Geng、Shuchang Liu、Zuohui Fu、Yingqiang Ge 和 Yongfeng Zhang。2022。作为语言处理的推荐（RLP）：统一的预训练、个性化提示与预测范式（p5）。载于*ACM*推荐系统会议论文集（*RecSys*）。299–315。[9] Wang-Cheng Kang 和 Julian McAuley。2018。自注意力序列推荐。载于*IEEE*国际数据挖掘会议论文集（*ICDM*）。197–206。[10] Dejiang Kong 和 Fei Wu。2018。HST-LSTM：一种用于位置预测的分层时空长短期记忆网络。载于国际人工智能联合会议论文集（*IJCAI*）。2341–2347。[11] Ranzhen Li、Yanyan Shen 和 Yanmin Zhu。2018。结合时间与多层次上下文注意力的下一个兴趣点推荐。载于*IEEE*国际数据挖掘会议论文集（*ICDM*）。1110–1115。[12] Defu Lian、Yongji Wu、Yong Ge、Xing Xie 和 Enhong Chen。2020。地理感知序列位置推荐。载于*ACM*知识发现与数据挖掘国际会议论文集（*KDD*）。2009–2019。[13] Defu Lian、Cong Zhao、Xing Xie、Guangzhong Sun、Enhong Chen 和 Yong Rui。2014年。GeoMF：兴趣点推荐的地理建模与矩阵分解联合方法。载于*ACM*知识发现与数据挖掘国际会议论文集（*KDD*）。831–840。[14] Pengfei Liu、Weizhe Yuan、Jinlan Fu、Zhengbao Jiang、Hiroaki Hayashi 和 Graham Neubig。2023。预训练、提示与预测：自然语言处理提示方法的系统综述。计算调查55, 9（2023），1–35。[15] Xiao Liu、Kaixuan Ji、Yicheng Fu、Zhengxiao Du、Zhilin Yang 和 Jie Tang。2021。*P-Tuning v2*：提示调优在不同规模和任务中普遍可媲美微调。技术报告 arXiv:2110.07602。[16] Yan Luo。2022。利用手机大数据刻画游客每日出行链。技术报告 arXiv:2205.14395。[17] Yingtao Luo、Qiang Liu 和 Zhaocheng Liu。2021。STAN：用于下一位置推荐的时空注意力网络。载于万维网会议论文集（*WWW*）。2177–2185。[18] Jarana Manotumruksa、Craig Macdonald 和 Iadh Ounis。2018。用于情境感知场所推荐的上下文注意力循环架构。载于*ACM*国际信息检索研究与发展会议论文集（*SIGIR*）。555–564。[19] David Massimo 和 Francesco Ricci。2018。利用广义用户行为模型进行下一兴趣点推荐。载于*ACM*推荐系统会议论文集（*RecSys*）。402–406。[20] Qi Pi、Guorui Zhou、Yujing Zhang、Zhe Wang、Lejian Ren、Ying Fan、Xiaoqiang Zhu 和 Kun Gai。2020。基于搜索的用户兴趣建模与终身用于点击率预测的序列行为数据。载于*ACM*国际信息与知识管理会议（*CIKM*）论文集。2685–2692。[21] Chaoming Song、Tal Koren、Pu Wang 和 Albert-László Barabási。2010年。人类移动尺度特性的建模。《自然·物理》6，10（2010年），818–823。[22] Ke Sun、Tieyun Qian、Tong Chen、Yile Liang、Quoc Viet Hung Nguyen 和 Hongzhi Yin。2020年。接下来去哪里：为兴趣点推荐建模长期和短期用户偏好。载于*AAAI*人工智能会议（*AAAI*）论文集。214–221。[23] Jiaxi Tang 和 Ke Wang。2018年。通过卷积序列嵌入实现个性化Top-N序列推荐。载于*IEEE*国际数据挖掘会议（*ICDM*）论文集。565–573。[24] Waldo R. Tobler。1970年。模拟底特律地区城市增长的计算机电影。《经济地理》46（1970年），234–240。[25] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser 和 Illia Polosukhin。2017年。注意力机制即一切所需。载于神经信息处理系统进展（*NeurIPS*）。5998–6008。[26] Jingyuan Wang、Jiawei Jiang、Wenjun Jiang、Chao Li 和 Wayne Xin Zhao。2021年。LibCity：用于交通预测的开放库。载于国际地理信息系统进展会议（*SIGSPATIAL*）论文集。145–148。[27] Zhaobo Wang、Yanmin Zhu、Qiaomei Zhang、Haobing Liu、Chunyang Wang 和 Tong Liu。2022年。用于下一兴趣点推荐的图增强时空网络。《*ACM*数据知识发现汇刊》16，6（2022年），1–21。[28] Shu Wu、Yuyuan Tang、Yanqiao Zhu、Liang Wang、Xing Xie 和 Tieniu Tan。2019年。基于图神经网络的会话推荐。载于*AAAI*人工智能会议（*AAAI*）论文集。346–353。[29] Min Xie、Hongzhi Yin、Hao Wang、Fanjiang Xu、Weitong Chen 和 Sen Wang。2016年。学习基于图的兴趣点嵌入以实现基于位置的推荐。载于*ACM*国际信息与知识管理会议（*CIKM*）论文集。15–24。[30] Hao Xue、Flora Salim、Yongli Ren 和 Nuria Oliver。2021年。MobTCast：利用辅助轨迹预测实现人类移动预测。载于神经信息处理系统进展（*NeurIPS*）。30380–30391。[31] Song Yang、Jiamou Liu 和 Kaiqi Zhao。2022年。GETNext：由轨迹流图增强的下一兴趣点推荐Transformer。载于*ACM*国际信息检索研究与发展会议（*SIGIR*）论文集。1144–1153。[32] Feng Yu、Yanqiao Zhu、Qiang Liu、Shu Wu、Liang Wang 和 Tieniu Tan。2020年。TAGNN：用于基于会话推荐的目标注意力图神经网络。载于*ACM*国际信息检索研究与发展会议（*SIGIR*）论文集。1921–1924。[33] Quan Yuan、Gao Cong 和 Aixin Sun。2014年。结合地理和时间影响的基于图的兴趣点推荐。载于*ACM*国际信息与知识管理会议（*CIKM*）论文集。659–668。[34] Yuyu Zhang、Hanjun Dai、Chang Xu、Jun Feng、Taifeng Wang、Jiang Bian、Bin Wang 和 Tie-Yan Liu。2014年。利用循环神经网络进行赞助搜索序列点击预测。载于*AAAI*人工智能会议（*AAAI*）论文集。1369–1375。[35] Vincent W Zheng、Yu Zheng、Xing Xie 和 Qiang Yang。2010年。利用GPS历史数据进行协同位置与活动推荐。载于万维网会议（*WWW*）论文集。1029–1038。[36] Yu Zhu、Hao Li、Yikang Liao、Beidou Wang、Ziyu Guan、Haifeng Liu 和 Deng Cai。2017年。接下来做什么：利用Time-LSTM对用户行为建模。。载于国际人工智能联合会议（*IJCAI*）论文集。3602–3608。