GETNext：用于下一兴趣点推荐的轨迹流图增强型 Transformer

Song Yang 奥克兰大学 奥克兰，新西兰
syan382@aucklanduni.ac.nz

Jiamou Liu 奥克兰大学 奥克兰，新西兰
jiamou.liu@auckland.ac.nz

Kaiqi Zhao 奥克兰大学 奥克兰，新西兰
kaiqi.zhao@auckland.ac.nz

arXiv:2303.04741v1 [cs.IR] 3 Mar 2023

摘要

下一兴趣点推荐旨在根据用户当前状态和历史信息，预测其近期的未来移动轨迹，这对用户和服务提供商都具有重要价值。然而，这一问题显然十分复杂，因为需要综合考虑多种数据趋势，包括空间位置、时间上下文、用户偏好等。大多数现有研究将下一兴趣点推荐视为序列预测问题，却忽略了其他用户提供的协同信号。为此，我们提出一种与用户无关的全局轨迹流动图，以及一种新颖的图增强 Transformer 模型（GETNext），以更充分地利用广泛的协同信号，从而更准确地预测下一兴趣点，同时缓解冷启动问题。GETNext 将全局转移模式、用户的一般偏好、时空上下文以及时间感知的类别嵌入整合到 Transformer 模型中，以预测用户未来的移动轨迹。凭借这一设计，我们的模型大幅优于当前最先进的方法，同时也为涉及时空因素的推荐问题中的冷启动挑战提供了启示。

CCS 概念

• **信息系统** → **推荐系统**；• **计算方法** → **神经网络**。

关键词

下一兴趣点推荐；图神经网络；Transformer

ACM 引用格式：

Song Yang、Jiamou Liu 和 Kaiqi Zhao。2022年。GETNext：用于下一兴趣点推荐的轨迹流图增强型 Transformer。载于第*45*届国际 *ACM* 信息检索研究与开发会议论文集（*SIGIR '22*），*2022*年7月*11*日至*15*日，西班牙马德里。ACM，纽约，NY，美国，10页。https://doi.org/10.1145/3477495.3531983

1 引言

近年来，随着支持 GPS 的移动设备日益普及，基于位置的服务（LBS）取得了显著进展。Foursquare、Gowalla 或 Yelp 等服务提供商使用户能够分享他们在兴趣点（POI）的体验、攻略和动态。因此，大量时空数据不断积累，推动下一兴趣点推荐系统日益强大 [5]。此类系统旨在根据用户当前和历史的足迹（即签到）预测他们将要访问的下一个兴趣点。这不仅有助于用户更好地探索周边环境，还能帮助企业改进广告策略 [8]。

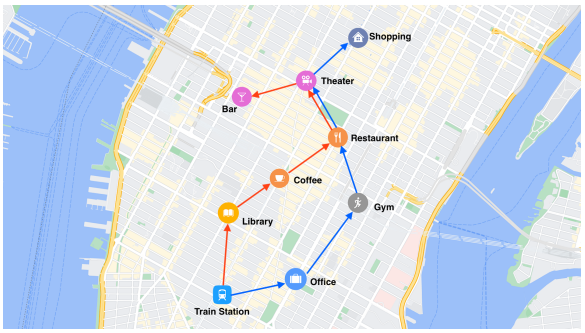


图1：纽约市曼哈顿两位用户在不同日期的两条轨迹（红色和蓝色）共享同一片段（从餐厅到剧院）

由于用户轨迹中的（短期）时间模式为预测用户未来的移动提供了重要线索，主流方法将下一兴趣点推荐视为序列预测任务。因此，该任务中最先进的模型采用各种形式的循环神经网络（RNN）来编码时空信息 [14, 19, 22, 23, 30, 31]。然而，这种方法存在三个局限。首先，与长轨迹相比，模型在短轨迹上的性能会显著下降。这是因为短轨迹提供的时空上下文有限。其次，对于签到过少量兴趣点的非活跃用户，推荐准确率往往较低。冷启动问题普遍存在于现实世界的推荐系统中。第三，兴趣点类别通常呈现出很强的时间相关性。例如，如图3所示，与午夜相比，火车站在高峰时段的签到频率显著更高。然而，现有模型未能建立时间与兴趣点类别之间的联系。尽管由于数据稀缺问题的固有存在，解决前两个问题并没有万全之策，但我们认为，利用其他用户的集体信息可以在一定程度上缓解这些局限。具体而言，个体往往会共享某些签到序列片段，从而形成集体轨迹流。例如，两位用户去了同一家餐厅和同一家

允许为个人或课堂使用而免费制作本作品全部或部分内容的数字或纸质副本，前提是这些副本不得用于营利或商业利益，也不得以营利或商业利益为目的进行复制或传播，且副本必须保留本声明及首页的完整引用信息。本作品中由 ACM 以外的其他方拥有版权的部分必须遵守相应版权规定。注明出处后可进行摘要引用。若要以其他方式复制、转载、发布到服务器或分发至列表，须事先取得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。*SIGIR '22*，*2022*年7月*11*日至*15*日，马德里，西班牙© 2022年 计算机协会。ACM ISBN 978-1-4503-8732-3/22/07...$15.00 https://doi.org/10.1145/3477495.3531983

电影院，且顺序相同，但日期不同，如图1所示。这些轨迹流可能为用户的通用移动模式提供重要线索，有助于解决短轨迹和非活跃用户带来的问题。然而，在下一兴趣点推荐中利用这些轨迹流并非易事，因为我们需要回答以下问题：**(1)** 如何聚合签到序列中的信息，以形成全局轨迹流模式的统一表示？**(2)** 除这些轨迹流之外，如何保留类别信息和用户偏好等重要的时空上下文信息？以及**(3)** 如何在下一兴趣点推荐中利用上述所有信息，在通用移动模式与个性化需求之间取得平衡？

对于**问题（1）**，我们构建了一种新颖的（与用户无关的）轨迹流图，用单一图结构概括兴趣点之间的轨迹以及每个兴趣点的特征。具体而言，该图中的节点是兴趣点，其属性包括地理位置、类别和签到次数。如果两个兴趣点在同一条签到轨迹中被连续访问，就用一条有向边连接它们，边权重则表示它们的共同访问频率。与现有兴趣点和商品推荐模型中用于建模用户与物品之间相关性的图不同，该轨迹流图捕捉的是兴趣点之间的转移影响。随后，为了利用轨迹流图中的集体信息，我们采用图卷积网络（GCN），将兴趣点嵌入能够保留兴趣点之间全局转移的潜在空间。具体来说，GCN通过聚合某个节点入邻居的嵌入来更新该节点的嵌入。这样，轨迹流图中每个兴趣点的嵌入都会受到其前序节点的影响，从而充分保留全局转移信息。因此，即使不知道用户的历史签到记录，我们仍然可以推荐当前兴趣点最可能的出邻居。全局轨迹流图背后的基本假设是，推荐热门的下一兴趣点或许不是最佳选择，但肯定胜过随机猜测。具体而言，使用全局轨迹流图应能提高相对较大的 k 下的top-k 准确率。参见第4.2节。

对于**问题（2）**，轨迹流图主要捕捉与用户无关的兴趣点转移模式。与此同时，用户偏好和时空上下文对于个性化推荐至关重要。长期偏好衡量用户的一般偏好，例如偏爱某家餐厅或电影院的倾向。用户的近期签到反映了短期偏好，并提供了更具体的时空上下文。因此，我们利用嵌入层捕捉用户的一般偏好，利用兴趣点类别嵌入表示兴趣点类别，并使用Time2Vec 模型描述时间嵌入。此外，兴趣点类别通常与时间密切相关。为了将两者结合起来，我们融合类别与时间嵌入，并将其输入融合模块，以生成时间感知类别上下文嵌入。详见第4.3节。

综上所述，我们提出了一种图增强 Transformer 框架 GETNext [1]，将通用移动模式、用户的一般偏好、短期时空上下文和时间感知类别嵌入统一起来，以预测用户接下来访问的兴趣点。与循环神经网络和长短期记忆网络相比，Transformer 可以利用自注意力机制，直接从输入轨迹中学习每次签到对最终推荐的贡献。换言之，模型可以区分不同签到所包含的信息量，并同时聚合轨迹中的所有签到进行预测，从而取得更优的性能。在 GETNext 中，我们采用 Transformer 编码器和多个多层感知机（MLP）解码器，以整合兴趣点嵌入及其他个性化嵌入中编码的隐式全局流动模式。此外，通过学习得到的转移注意力图，也将全局流动模式显式地注入最终预测中。这一新颖的模型回答了上文的**问题（3）**。详见第 4.4 节。

为了验证我们提出的模型，我们在广为人知的真实世界数据集上进行了一系列实验。实验结果表明，GETNext 能够显著优于现有最先进的方法。例如，在纽约市数据集上，Top-5准确率最高提升了 11%。详见第5节。下面列出我们工作的主要贡献：

(1) 我们提出一种全局轨迹流动图，用于对常见的访问顺序转移信息进行建模，并利用图卷积网络将其编码为兴趣点嵌入。**(2)** 我们设计了一种新颖的时间感知类别上下文嵌入，以捕捉兴趣点类别多样的时间模式。**(3)** 我们提出一种基于 Transformer 的框架，将全局转移模式、用户的一般偏好、用户的短期轨迹和时空上下文整合起来，用于下一兴趣点推荐。

2 相关工作

2.1 下一兴趣点推荐

与传统的兴趣点推荐相比，下一兴趣点推荐 [2] 更加关注近期轨迹的时间影响，以预测用户的下一步行动。早期研究采用了其他序列推荐任务中广泛使用的方法，例如马尔可夫链 [2, 26, 29]。例如，Cheng 等人 [2] 的开创性工作采用矩阵分解方法推荐下一兴趣点，并将个性化马尔可夫链嵌入其中（FPMC）[18]。类似地，Zhang 等人 [29]提出了一种加性马尔可夫链，用于建模序列转移影响。与此同时，其他研究探索了将常用的矩阵分解或度量嵌入技术应用于下一兴趣点推荐的可能性 [5, 15, 32]。总体而言，与深度神经网络模型相比，这些早期方法在序列数据建模能力方面相当有限。

近来，研究人员转向深度学习和先进的嵌入方法 [6]。研究人员提出了循环神经网络的变体，以捕捉时间动态和序列相关性 [14, 19, 22, 23, 30, 31] 2016, [14] 。

Liu 等人提出了时空循环神经网络（ST-RNN），将时空上下文融入循环神经网络层。具体而言，空间上下文由地理距离转移矩阵表示，而时间转移矩阵则编码时间上下文。此外，长短期记忆网络也被用于建模用户的长期和短期偏好。在 LSPL[22]和 PLSPL [23], 中，作者训练标准长短期记忆网络模型，用于

[1]代码见 https://github.com/songyangco/GETNext

短期轨迹挖掘，并使用通用嵌入层捕捉用户偏好。Zhao 等人 [31] 设计了一种新颖的长短期记忆网络单元，称为时空门控网络（STGN），它包含两个时间门控和两个距离门控，用于对短期和长期序列中的时间间隔与距离间隔进行建模。总体而言，上述所有工作都将下一兴趣点推荐视为序列预测任务。也有少数研究将注意力机制用于这一推荐任务，例如 DeepMove [4] 和 STAN [16]。DeepMove [4] 提出了一种注意力模型来捕捉多层次周期模式，并采用循环神经网络对序列转移进行建模，以生成最终推荐。STAN [16]采用自注意力机制提取非相邻的点对点交互，而序列模型无法做到这一点。然而，所有这些研究都忽视了利用用户一般移动模式的潜在益处。

2.2 基于位置的推荐中的图

本文旨在利用基于图的方法进行下一兴趣点推荐。基于图的方法——例如利用基于位置的社交网络（LBSN）的方法——为推荐任务提供了强大的范式，尤其适用于传统的（非序列）推荐任务。事实上， [28] 构建了一个地理时序影响感知图（GTAG），这是一个由兴趣点、会话节点和用户节点组成的三部图。然而，引入会话节点可能会导致图规模急剧膨胀。与 GTAG 不同，[24]构建了四个二部图，包括兴趣点-兴趣点、兴趣点-区域、兴趣点-时间和兴趣点-词语交互图，分别用于捕捉序列效应、地理影响、时间动态和语义特征。随后，作者将 LINE 网络嵌入模型 [20] 扩展至二部图，并使用条件概率及其他统计工具训练图嵌入。一项类似的研究是 GGLR [1], 基于图的地理潜在表示模型。GGLR 使用兴趣点-兴趣点图和兴趣点-用户图。在兴趣点-用户二部图中，如果用户的签到历史中曾访问过某个兴趣点，则用户节点与该兴趣点节点相连，旨在突出用户偏好。

我们指出，上述所有研究针对的都是传统的（非“下一兴趣点”）兴趣点推荐问题。据我们所知，尚无研究明确利用统一图结构进行下一兴趣点推荐。在一项近期研究中， [12], 对于给定签到序列中的每个兴趣点，作者从其他签到序列中随机采样其前序和后续签到，并将这些兴趣点纳入训练，以学习当前兴趣点的嵌入。其目标是捕捉兴趣点的局部（单跳）转移，因此该嵌入无法明确反映全局（多跳）转移模式。相比之下，我们的轨迹流图是一种统一图结构，展现了所有兴趣点之间的全局模式。从这个意义上说，我们的模型首次利用基于图的学习，在下一兴趣点推荐任务中编码兴趣点的一般转移信息。

3 问题定义

设 $U = \{u_1, u_2, \cdots u_M\}$ 为用户集合， $P = \{p_1, p_2, \cdots p_N\}$ 为兴趣点集合（例如特定的餐馆、酒店）， $T = \{t_1, t_2, \cdots t_K\}$ 表示时间戳集合，其中 M、 N、 K 均为正整数。每个兴趣点 $p \in P$ 表示为一个元组 $p = (lat, lon, cat, freq)$ 分别表示纬度、经度、类别和签到频率。现在我们定义本文中的几个关键概念。特别地， cat 取自一个固定的兴趣点类别 列表（例如“火车站”、“酒吧”）。

定义 3 1 $q = (u, p, t) \in U \times P \times T$（签到）。一次签到是一个元组 (u, p, t)，表示用户在时间戳访问过兴趣点 p。

用户 $u \in U$创建的所有签到活动构成一个签到序列 $Q_u = (q_u^1, q_u^2, q_u^3, \cdots)$，其中 q_u^i 表示第 i 条签到记录。将所有用户的签到序列记为 $Q_U = \{Q_{u_1}, Q_{u_2}, \cdots Q_{u_M}\}$.

作为数据预处理的一部分，我们将任意用户 u 的签到序列 Q_u 拆分为一组连续的轨迹，即 $S_u^1 \oplus S_u^2 \oplus \cdots$，其中 $\oplus$ 表示拼接。轨迹的长度可能不同，每条轨迹都包含在较短时间间隔（例如 24 小时）内的一系列签到。

下一兴趣点推荐 旨在通过学习当前轨迹和用户的所有历史签到记录，提供一份用户接下来可能会访问的兴趣点列表。更正式地说，给定一组历史轨迹 $\{S_u^i\}_{i \in \mathbb{N},\ u \in U}$，以及特定用户 $u_i \in U$的当前轨迹 $S_i = (q_1, q_2, \ldots, q_m)$，预测 u_i在接下来会访问的、可能性最高的 $q_{m+1}, q_{m+2}, \ldots, q_{m+k}$ 个兴趣点，其中 $k \geq 1$是一个较小的整数（通常为 $k = 1$）。

4 我们的方法：GETNEXT

4.1 模型结构概述

图2展示了我们提出的 GETNext 模型的整体设计。该模型融合了几个关键组件。首先，我们定义了轨迹流图，用于概括历史轨迹（见第4.2节）。轨迹流图通过两种方式影响推荐结果：

(1) 在轨迹流图上训练图神经网络（GNN），以生成兴趣点嵌入，编码用户在各个兴趣点上的通用移动模式，同时纳入兴趣点的类别、位置和签到频率。(2) 一个注意力模块，以轨迹流图的邻接矩阵和节点特征为输入，并生成转移注意力图，显式建模兴趣点之间的转移概率。

随后，我们定义了若干上下文模块，以获得用户嵌入、兴趣点类别嵌入和时间编码（通过 time2vector 模型）（见第4.3节）。

- 将相应轨迹中的用户嵌入与兴趣点嵌入相结合，以实现更好的个性化。
- 还将兴趣点类别嵌入与时间编码相结合，以捕捉用户对不同兴趣点类别的时间偏好（例如，高峰时段的火车站）。

上述方法使我们能够将签到记录中的用户、兴趣点类别、时间戳和兴趣点信息整合，生成一个签到嵌入 向量。随后，每条轨迹都可以编码为由此类签到嵌入组成的列表。接着，我们采用 Transformer 编码器和多层感知机（MLP）头来生成兴趣点预测。最后，通过残差连接，利用学习得到的转移注意力图调整预测的兴趣点。

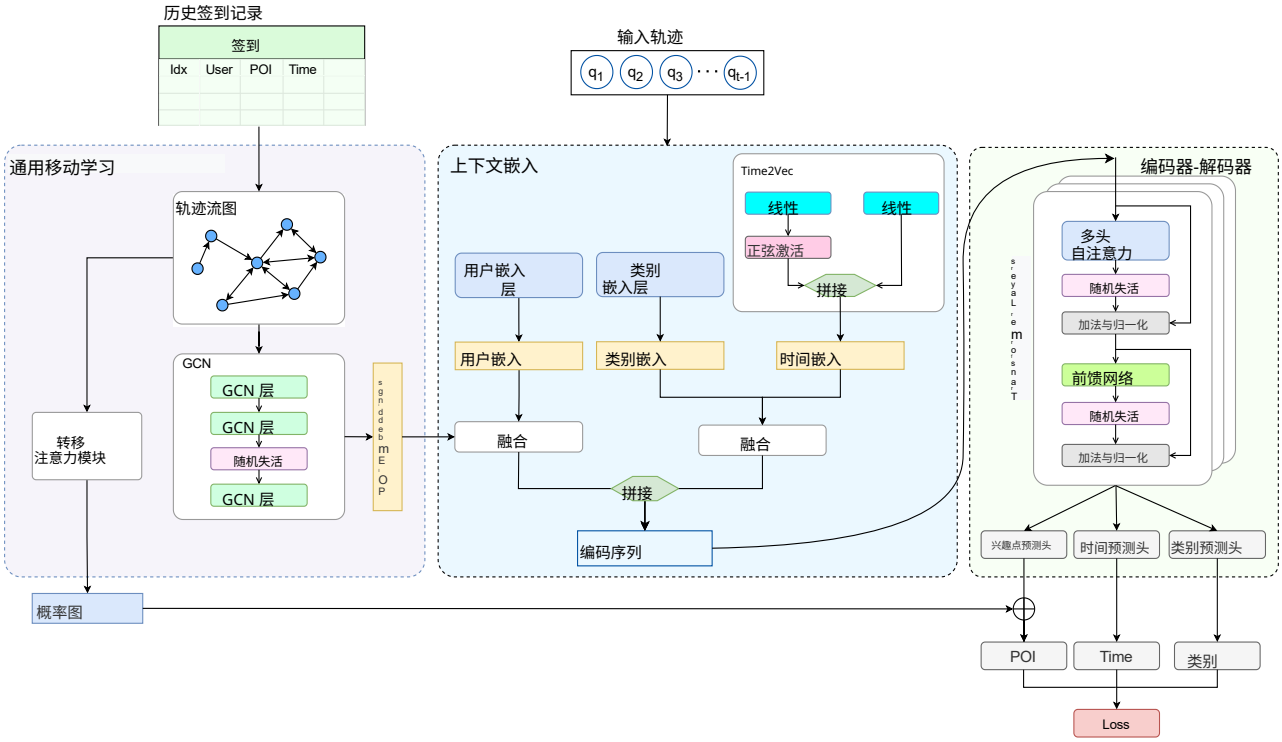


图2：所提出的GETNext模型概览

4.2 基于轨迹流图的学习

4.2.1 兴趣点嵌入。 我们观察到，不同个体可能共享某些相似的轨迹片段，同一个人也可能重复多次走过同一条轨迹。为了利用历史签到记录中的这些共同模式，我们构建了一个（与用户无关的）轨迹流图，以全局视图呈现用户在兴趣点之间的一般移动模式。

定义 4.1（轨迹流图）。给定历史轨迹集合$\mathcal{S} = \{S_u^i\}_{i \in \mathbb{N}, u \in U}$，轨迹流图是一个属性加权有向图$\mathcal{G} = (V, E, \ell, w)$，满足

- 节点集合为 $V = P$ 兴趣点集合，
- 每个 $p \in P$ 的属性 $\ell(p)$ 由 $(lat, lon, category, freq)$ 组成，其中 (lat, lon) 是 p 的坐标，$category$ 是 p 的类别，$freq$ 是 p 在 $\mathcal{S}$ 中的轨迹里出现的次数。

- 若 (p_1, p_2) 出现在 $\mathcal{S}$ 中的一条轨迹 S_u^i 里，即两者被连续访问，则存在一条从 p_1 到 p_2 的边。
- 任意边 (p_1, p_2) 的权重 $w(p_1, p_2)$ 等于 (p_1, p_2) 在 $\mathcal{S}$ 中的任意轨迹里出现的次数。

我们指出，轨迹流图不仅显示了用户在兴趣点之间的移动，也隐含地揭示了兴趣点之间的某种空间邻近性。直观而言，如果两个兴趣点相距较远，它们就不太可能被连续签到。

给定轨迹流图 G，下一步是学习兴趣点的向量化表示，以编码常见的兴趣点转移模式和兴趣点的属性。为此，我们使用图卷积网络（GCN）。为了充分利用 G 的拓扑信息，我们使用谱图卷积网络 [10]。具体而言，令 $\mathrm{A} \in \mathbb{R}^{N \times N}$ 表示 G 的邻接矩阵，我们首先计算归一化拉普拉斯矩阵：

$$\tilde{\mathrm{L}} = (\mathrm{D} + \mathrm{I}_N)^{-1}(\mathrm{A} + \mathrm{I}_N) \tag{1}$$

其中，D 是度矩阵，I_N 是 G 的单位矩阵。接下来，令 $\mathrm{H}^{(0)} = \mathrm{X} \in \mathbb{R}^{N \times C}$ 为输入节点特征矩阵。我们将 GCN 层之间的传播规则定义为

$$\mathrm{H}^{(l)} = \sigma\left(\tilde{\mathrm{L}}\mathrm{H}^{(l-1)}\mathrm{W}^{(l)} + b^{(l)}\right) \tag{2}$$

其中，$\mathrm{H}^{(l-1)}$ 表示任意 $l > 0$ 的第 l 层输入信号，$\mathrm{W}^{(l)} \in \mathbb{R}^{C \times \Omega}$ 表示第 l 层的模型权重矩阵，相应的偏置为 $b^{(l)} \in \mathbb{R}^{C \times \Omega}$，而 σ 是用于引入非线性的带泄漏的 ReLU 激活函数（泄漏率为 0.2）。

从空间角度来看，在每次迭代中，GCN 层都会通过聚合节点自身的嵌入及其邻域信息来更新该节点的嵌入。我们堆叠 l^* 个 GCN 层，以增强模型的表达能力。在最后一层之前使用随机失活。图卷积网络模块的输出可表示为：

$$\mathrm{e_P} = \tilde{\mathrm{L}}\,\mathrm{H}^{(l^*)}\mathrm{W}^{(l^*+1)} + b^{(l^*+1)} \in \mathbb{R}^{N \times \Omega} \tag{3}$$

最后，兴趣点 p_i 的嵌入 e_{p_i} 是矩阵 $\mathrm{e_P}$ 的第 i 行。简单来说，兴趣点 p 的嵌入表示 p 在所有用户历史轨迹中的位置，因此能够捕捉 p 处的通用移动模式。随后，它将被输入下游 Transformer，以对用户的访问行为进行建模。请注意

即使当前轨迹较短，兴趣点嵌入仍能为预测模型提供丰富的信息。

4.2.2 转移注意力图。从图 G 学到的兴趣点嵌入只能隐式地捕捉通用移动模式。为了增强集体信号的影响，我们提出一种新颖的转移注意力图，以显式地建模从一个兴趣点到另一个兴趣点的转移概率。如上所述，这些转移概率将用于调整最终预测。给定输入节点特征和 G，我们计算注意力图 Φ，如下所示：

$$\Phi_1 = (\mathrm{X} \times \mathrm{W}_1) \times \mathrm{a}_1 \in \mathbb{R}^{N\times 1} \tag{4}$$

$$\Phi_2 = (\mathrm{X} \times \mathrm{W}_2) \times \mathrm{a}_2 \in \mathbb{R}^{N\times 1} \tag{5}$$

$$\Phi = (\Phi_1 \times 1^\top + 1 \times \Phi_2^\top) \odot (\tilde{\mathrm{L}} + J_N) \in \mathbb{R}^{N\times N} \tag{6}$$

其中，W_1,$\mathrm{W}_2 \in \mathbb{R}^{C\times h}$ 是两个可训练的特征变换矩阵；$\mathrm{a}_1, \mathrm{a}_2 \in \mathbb{R}^h$ 是两个可学习向量，用于通过广播加法操作构造注意力矩阵；$\mathbf{1}$ 是形状为 $\mathbb{R}^{N\times 1}$ 的全 1 向量；J_N 是全 1 矩阵，$\odot$ 表示逐元素乘法。我们将归一化拉普拉斯矩阵 L 的取值范围从 $[0, 1]$ 平移到 $[1, 2]$，以避免出现零值。

转移注意力图 Φ 的第 i 行表示从兴趣点 p_i 移动到各个兴趣点的（未归一化）概率。给定当前轨迹中的最后一个兴趣点，我们查询 Φ 中对应行所存储的转移概率，并使用这些概率调整后续 Transformer 模块生成的推荐结果。

4.3 上下文嵌入模块

时空上下文和用户偏好是个性化下一兴趣点推荐的关键因素 [6, 14, 31]。接下来，我们将介绍上下文嵌入模块，用于融合用户嵌入、兴趣点嵌入、兴趣点类别嵌入和时间编码等上下文信息。

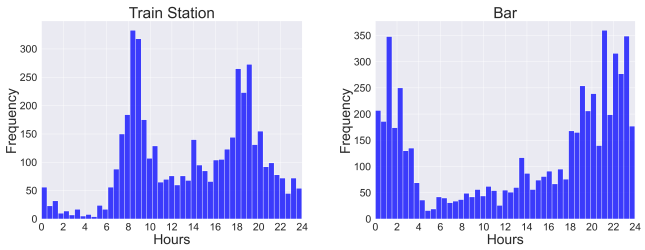
4.3.1 兴趣点-用户嵌入融合。兴趣点嵌入由轨迹流图学习得到，不包含用户特定模式。为了捕捉特定用户 u 的一般行为，我们训练一个嵌入层，将每位用户映射为低维向量。每位用户的嵌入都是根据其历史签到序列学习得到的。形式上，用户嵌入 u 为

$$\mathrm{e}_u = f_{\text{embed}}(u) \in \mathbb{R}^{\Omega}. \tag{7}$$

为了构建每条签到活动的表示，一种直接的方法是将兴趣点嵌入与用户嵌入拼接起来。我们将拼接后的向量输入全连接层，以微调融合后的嵌入并增强其表示能力。输出可表示为

$$\mathrm{e}_{p,u} = \sigma(\mathrm{w}_{p,u}[\mathrm{e}_p; \mathrm{e}_u] + b_{p,u}) \in \mathbb{R}^{\Omega\times 2}, \tag{8}$$

其中，p、u 和 b_p、u 分别表示权重向量和偏置，$[\cdot;\cdot]$ 表示拼接。输出嵌入的维度是兴趣点嵌入或用户嵌入的两倍。换句话说，融合后嵌入向量的大小保持不变。



(a) “火车站” **(b) “酒吧”**

图 3：纽约市数据集中两种兴趣点类别（“火车站”和“酒吧”）的平均每小时签到频率

4.3.2 时间-类别嵌入融合。用户的访问行为自然具有时间依赖性。例如，图3展示了一天中两种兴趣点类别的签到频率。“火车站”在高峰时段（约上午8点和下午6点）出现两个明显的峰值。相反，“酒吧”呈现出完全相反的模式，大多数签到活动发生在下午6点之后。这些观察促使我们在进行下一兴趣点推荐时考虑兴趣点类别的时间模式。例如，上午8点时，应推荐火车站，而不是酒吧。

值得注意的是，单个兴趣点的签到也可能呈现某些时间模式。然而，由于签到数量稀少且数据中存在噪声，单个兴趣点的时间模式远不如类别层面的模式清晰和稳定。以纽约市数据集为例，该数据集包含 22.7 万条签到记录、3.8 万个兴趣点和 400 个类别。平均而言，每个兴趣点的签到次数不到 6 次。相比之下，每个类别的签到次数超过 570 次。因此，我们研究兴趣点类别的时间模式，而非单个兴趣点的时间模式。

与兴趣点-用户嵌入融合模块类似，我们首先对兴趣点类别和时间进行编码。我们采用先进的时间编码模型 time2vector [9]。具体而言，我们将一天的 24 小时划分为 48 个时隙，每个时隙为 30 分钟。我们将一个（标量）时间值映射到其中一个时间时隙。时间时隙 t 的嵌入记为 e_t，它是一个长度为 $k+1$ 的向量。第 i 个元素定义如下：

$$\mathrm{e}_t[i] = \begin{cases} \omega_i t + \varphi_i, & \text{if } i = 0. \\ \sin(\omega_i t + \varphi_i), & \text{if } 1 \le i \le k. \end{cases} \tag{9}$$

其中，ω和 φ是可学习参数。正弦激活函数用于捕捉周期模式。

我们为兴趣点类别采用另一个嵌入层。类别 c 的嵌入为

$$\mathrm{e}_c = f_{\text{embed}}(c) \in \mathbb{R}^{\Psi}, \tag{10}$$

其中，Ψ 是嵌入维度。

接下来，我们通过一个全连接层融合时间嵌入 e_t 和类别嵌入 e_c，形成单一的类别-时间表示：

$$\mathrm{e}_{c,t} = \sigma(\mathrm{w}_{c,t}[\mathrm{e}_t; \mathrm{e}_c] + b_{c,t}) \in \mathbb{R}^{\Psi\times 2}, \tag{11}$$

其中，$\mathrm{w}_{c,\ t}$ 是可学习权重向量，$b_{c,\ t}$ 是偏置。

最后，签到 $q = p,\ u,\ t$ 的嵌入（其中兴趣点 p的类别为 c）是 $\mathrm{e}_q = [\mathrm{e}_{p,\ u}; \mathrm{e}_{c,\ t}]$的拼接。因此，每条输入轨迹（$q_1, \ldots, q_s$）都表示为一组签到

嵌入（$\mathbf{e}_{q_1}, \ldots, \mathbf{e}_{q_s}$）。我们将把编码序列输入 Transformer 编码器。

4.4 Transformer 编码器和多层感知机解码器

4.4.1 Transformer 编码器。Transformer 是一种纯粹基于注意力机制的编码器-解码器网络，最初由 Vaswani等人 [21] 于 2017 年提出。由于其计算效率出众，且相较于传统模型性能优异，Transformer 已成为自然语言处理领域的首选范式，近年来也应用于图像识别 [3], 、视频理解 [17], 和时间序列分析 [13]。

我们采用 Transformer 进行下一兴趣点推荐，因为这本质上是一项序列预测任务。由于目标是根据签到序列仅预测紧接着的下一个兴趣点，因此我们只采用 Transformer 编码器，并在其后接入若干多层感知机头（不使用 Transformer 解码器）。对于编码器，我们堆叠若干标准 Transformer 编码器层，并加入位置编码。每层由一个多头自注意力模块和一个逐位置全连接网络组成。两个模块均采用残差连接和归一化。

形式化地说，给定输入轨迹 $S_u = (q_u^1, q_u^2, \ldots\ldots, q_u^k)$，为了预测下一次签到活动 q_u^{k+1}中的兴趣点，我们首先按照第 4.3 节中的定义，获取每条历史签到记录 $q_u^i, 1 \le i \le k$的签到嵌入。然后，将这些签到嵌入堆叠起来，构成第一个编码器层的输入张量，记为 $\mathbf{X}^{[0]} \in \mathbb{R}^{k \times d}$，其中 d表示每条签到记录的嵌入维度。特别地，d等于 $\mathbf{e}_q$的长度，即 $d = 2 \times (\Omega + \Psi)$。值得注意的是，为便于残差连接和层堆叠，所有编码器层中的嵌入长度 d均保持不变。换言之，忽略填充后，编码器层 l 的输出形状与输入形状一致，即 $\mathbf{X}^{[l+1]} \in \mathbb{R}^{k \times d}$。

对于第 l 层，输入 $\mathbf{X}^{[l]} \in \mathbb{R}^{k \times d}$首先经过多头自注意力模块（在我们的实验中，注意力头的数量为 2）进行变换。对于第一个注意力头，其输出为:

$$S = \mathcal{X}^{[l]} \boldsymbol{W}_q (\mathcal{X}^{[l]} \boldsymbol{W}_k)^T \in \mathbb{R}^{k \times k} \tag{12}$$

$$S'_{i,j} = \frac{\exp(S_{i,j})}{\sum_{j=1}^{d} \exp(S_{i,j})} \tag{13}$$

$$\text{head}_1 = S' \mathcal{X}^{[l]} \boldsymbol{W}_v \in \mathbb{R}^{k \times d/h}, \tag{14}$$

其中，$\boldsymbol{W}_q \in \mathbb{R}^{d \times d/h}$、$\boldsymbol{W}_k \in \mathbb{R}^{d \times d/h}$和 $\boldsymbol{W}_v \in \mathbb{R}^{d \times d/h}$分别是对应于“查询”、“键”和“值”的可学习权重矩阵。这里，我们在公式中忽略偏置。点积注意力，即 $S_{i,j}$表示第 i 个与第 j 个签到活动之间的相关性。接下来，应用 Softmax函数以确保注意力权重之和为 1。将输入数据通过 $\boldsymbol{W}_v$投影到输出特征空间后，我们使用学习到的注意力矩阵调整每条签到记录的贡献。最后，我们将不同的注意力头堆叠起来，并采用另一个线性变换 $\boldsymbol{W}_o \in \mathbb{R}^{d \times d}$，合并来自不同注意力空间的表征:

$$\text{Multihead}(\mathcal{X}^{[l]}) = [\text{head}_1; \cdots; \text{head}_h] \times \boldsymbol{W}_o \in \mathbb{R}^{k \times d}. \tag{15}$$

此外，层归一化和残差连接被应用于注意力模块。因此，注意力模块的最终输出可表示为

$$\mathcal{X}_{\text{attn}}^{[l]} = \text{LayerNorm}\left(\mathcal{X}^{[l]} + \text{Multihead}(\mathcal{X}^{[l]})\right). \tag{16}$$

在每个编码器层中，注意力模型之后都会接入一个全连接网络。将多头注意力模块的输出记为 $\mathbf{X}^{[l]}\text{attn}$。全连接网络可表示为

$$\mathcal{X}_{\text{FC}}^{[l]} = \text{ReLU}(\mathbf{W}_1 \mathcal{X}_{\text{attn}}^{[l]} + b_1)\mathbf{W}_2 + b_2 \in \mathbb{R}^{k \times d}, \tag{17}$$

其中，$\mathbf{W}_1, \mathbf{W}_2$是可训练的权重矩阵，$b_1$、$b_2$是偏置。同样，第 l 个编码器层的输出为

$$\mathcal{X}^{[l+1]} = \text{LayerNorm}(\mathcal{X}_{\text{attn}}^{[l]} + \mathcal{X}_{\text{FC}}^{[l]}) \in \mathbb{R}^{k \times d}. \tag{18}$$

4.4.2 多层感知机解码器。Transformer 编码器层将输入轨迹签到嵌入中的有用信息提炼到特征空间中。为了预测用户的下一步移动，我们用多个多层感知机（MLP）解码器替换 Transformer 解码器。具体而言，我们使用三个多层感知机头，分别预测下一个兴趣点、访问时间和兴趣点类别。设编码器的输出为 $\mathbf{X}^{[l^*]}$，则多层感知机头可表示为

$$\hat{\mathbf{Y}}_{\text{poi}} = \mathcal{X}^{[l^*]} \mathbf{W}_{\text{poi}} + b_{\text{poi}} \tag{19}$$

$$\hat{\mathbf{Y}}_{\text{time}} = \mathcal{X}^{[l^*]} \mathbf{W}_{\text{time}} + b_{\text{time}} \tag{20}$$

$$\hat{\mathbf{Y}}_{\text{cat}} = \mathcal{X}^{[l^*]} \mathbf{W}_{\text{cat}} + b_{\text{cat}} \tag{21}$$

其中，$\mathbf{W}\text{poi} \in \mathbb{R}^{d \times N}$、$\mathbf{W}\text{time} \in \mathbb{R}^{d \times 1}$、$\mathbf{W}\text{cat} \in \mathbb{R}^{d \times \Gamma}$ 是多层感知机中的权重，Γ 表示兴趣点类别的数量。对于兴趣点预测头 $\hat{}\mathbf{Y}\text{poi} \in \mathbb{R}^{k \times N}$的输出，我们只关注最后一行，它对应于对未来移动的兴趣点推荐。此外，我们将该兴趣点推荐与第 4.2.2节中定义的转移注意力图相结合。最终的推荐为

$$\hat{y}_{\text{poi}} = \hat{\mathbf{Y}}_{\text{poi}}^{(k\cdot)} + \Phi^{(p_k \cdot)} \in \mathbb{R}^{1 \times N}, \tag{22}$$

其中，$\hat{}\mathbf{Y}(k\cdot)\text{poi}$ 表示 $\hat{}\mathbf{Y}poi$ 的第 k 行，$\Phi(p_k\cdot)$ 是转移图 Φ 的第 p_k 行，p_k 表示签到 q_u^k 对应的兴趣点。

如上所示，除了兴趣点预测头之外，我们还添加了时间预测头和类别预测头。我们在预测兴趣点的同时预测下一次签到时间，主要原因如下：两次签到之间的时间间隔波动很大（例如，从半小时到长达 6 小时不等）。这是合理的，因为用户在不同兴趣点停留的时间长短不一，或者会忘记记录某些签到。然而，这种波动会对预测产生重大影响。事实上，对于下午 5 点的用户而言，接下来 1 小时和接下来 5 小时的推荐应有所不同。因此，我们不仅推荐下一个兴趣点，也预测下一次签到时间，并使用时间预测头对时间建模进行校准。此外，类别预测头用于调节下一兴趣点预测，因为预测下一兴趣点类别比精确预测兴趣点更容易。

4.4.3 损失。训练时会考虑解码器中的所有多层感知机头，其中时间和类别预测头的输出作为正则化项。换言之，我们计算所有多层感知机头的损失加权和。兴趣点和兴趣点类别预测采用交叉熵作为损失函数。时间预测则采用均方误差（MSE）作为损失函数。

此外，由于我们将一天内的 24 小时归一化到 $[0, 1]$, ，时间损失的数值明显小于另外两项损失。为使梯度幅度与其他损失保持平衡，我们将时间损失项放大 10 倍，即最终损失为

$$\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{poi}} + 10 \times \mathcal{L}_{\text{time}} + \mathcal{L}_{\text{cat}}. \tag{23}$$

5 实验

本节将在真实世界数据集上评估我们提出的模型。

5.1 实验设置

5.1.1 数据集。我们在从基于位置的服务平台收集的三个公开数据集上进行实验：FourSquare-NYC [25], FourSquare-TKY [25], 和 Gowalla-CA [27]。FourSquare-NYC 收集于 2012 年 4 月至 2013 年 2 月，地点为纽约市；FourSquare-TKY 则在同期收集于东京。Gowalla-CA 包含来自 Gowalla [27] 的签到记录，时间跨度为 2009 年 2 月至 2010 年 10 月，地点涵盖加利福尼亚州和内华达州。每条记录包含用户、兴趣点、兴趣点类别、GPS 坐标和时间戳。对于这三个数据集，我们剔除了签到记录少于 10 条的不热门兴趣点，也过滤掉签到历史少于 10 条的用户。随后，我们以 24 小时为间隔，将用户的完整签到序列划分为轨迹。在少数情况下，轨迹仅包含一次签到；我们将这些情况从数据集中剔除。接着，我们按时间顺序将数据集划分为训练集、验证集和测试集。前 80% 的签到记录作为训练集，用于构建轨迹流图 G；中间 10% 作为验证集，剩余 10% 构成测试集。如果用户或兴趣点未在训练集中出现，却在测试集中出现，那么在评估预测性能时，我们会跳过该用户或兴趣点。数据集的关键统计信息见表 1。

表 1：数据集统计

	#用户	#poi	#cat	#签到	#轨迹
NYC	1,075	5,099	318	104,074	14,160
TKY	2,281	7,844	291	361,430	44,692
CA	4,318	9,923	301	250,780	32,920

5.1.2 评估指标。我们计算准确率@k（Acc@k）和平均倒数排名（MRR），这两项是推荐系统中常用的指标。准确率@k 表示真实兴趣点是否出现在排名前k 的推荐兴趣点中。由于 Acc@k 将排名前k的推荐结果视为无序列表，忽略正确预测的排序，因此我们采用 MRR 来衡量正确推荐的兴趣点在有序结果列表中的位置。给定一个包含 m 个样本（轨迹）的数据集，定义

$$\text{Acc@}k = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(rank \le k)$$

$$\text{MRR} = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{rank}$$

其中，$\mathbb{1}$ 是指示函数。条件为真时返回 1，否则返回 0。排名表示真实下一个兴趣点在推荐排序列表中的排名。一般而言，对于所有这些指标，值越大，性能越好。

5.1.3 基线模型。我们采用以下基线模型 .

• MF [11] 是许多推荐问题中的一种经典方法。它通过矩阵分解学习用户和兴趣点的潜在表示。

• FPMC [18] 将矩阵分解与马尔可夫链结合起来，同时建模用户的长期偏好和序列行为。

• 长短期记忆网络 [7] 是一种用于处理序列数据的循环神经网络模型变体。与标准循环神经网络模型相比，长短期记忆网络能够建模短期和长期序列模式。

• PRME [5] 提出了一种名为个性化排序度量嵌入的成对嵌入方法，用于捕捉用户偏好以及兴趣点之间的序列转移。

• ST-RNN [14]采用时间和距离转移矩阵来建模局部时空上下文，并使用循环神经网络捕捉用户的序列模式。

• STGN [31] 通过添加空间门和时间门扩展了传统的长短期记忆网络，以捕捉用户在空间和时间维度上的偏好。

• STGCN [31] 是 STGN 的改进版本，采用了耦合输入门和遗忘门。

• PLSPL [23] 通过注意力机制学习用户的长期偏好，并使用长短期记忆网络学习短期偏好，再通过个性化线性层将二者结合起来。

• STAN [16] 利用轨迹中签到记录的时空信息，并通过自注意力层捕捉非相邻签到记录之间的点对点交互。

5.1.4 实验设置。 我们使用 PyTorch 框架开发了模型，并在以下硬件平台上进行了实验（CPU：AMD Ryzen 9 5900X, GPU：NVIDIA GeForce RTX 3090）。模型的关键超参数设置如下。兴趣点和用户的嵌入维度均为 $\Omega = 128$。时间和兴趣点类别的嵌入长度为 $\Psi = 32$。GCN 模型包含三个隐藏层，通道数分别为 32、64 和 128。转移注意力模块将输入节点特征转换为 128 维向量。对于 Transformer，我们堆叠了两个编码器层。Transformer 编码器层中前馈网络的维度为 1024，多头注意力模块采用两个注意力头。此外，我们使用 Adam优化器，学习率为 1e-3，权重衰减率为 5e-4。GCN 模型和 Transformer 编码器均启用了随机失活，比例为 0.3。另一个重要参数是时间损失的权重 α，我们将其设为 10，以匹配兴趣点损失和类别损失的尺度。我们在三个数据集上使用相同的设置，并以批量大小 20 对每个模型训练 200 轮。

5.2 实验结果

表 2 展示了我们的模型与基线模型在三个数据集上的性能比较。我们报告了 Top-1、Top-5、Top-10、Top-20 准确率和平均倒数排名。总体而言，所有模型在纽约市和东京数据集上的表现都优于在加利福尼亚数据集上的表现。主要原因是，纽约市和东京的兴趣点被限制在相对较小的区域内（纽约

表 2：三个数据集上的 Acc@k 准确率和平均倒数排名性能比较

	NYC					TKY					CA				
	Acc@1	Acc@5	Acc@10	Acc@20	平均倒数排名	Acc@1	Acc@5	Acc@10	Acc@20	平均倒数排名	Acc@1	Acc@5	Acc@10	Acc@20	平均倒数排名
MF	0.0368	0.0961	0.1522	0.2375	0.0672	0.0241	0.0701	0.1267	0.1845	0.4861	0.0110	0.0442	0.0723	0.1190	0.0342
FPMC	0.1003	0.2126	0.2970	0.3323	0.1701	0.0814	0.2045	0.2746	0.3450	0.1344	0.0383	0.0702	0.1159	0.1682	0.0911
LSTM	0.1305	0.2719	0.3283	0.3568	0.1857	0.1335	0.2728	0.3277	0.3598	0.1834	0.0665	0.1306	0.1784	0.2211	0.1201
PRME	0.1159	0.2236	0.3105	0.3643	0.1712	0.1052	0.2278	0.2944	0.3560	0.1786	0.0521	0.1034	0.1425	0.1954	0.1002
时空循环神经网络	0.1483	0.2923	0.3622	0.4502	0.2198	0.1409	0.3022	0.3577	0.4753	0.2212	0.0799	0.1423	0.1940	0.2477	0.1429
STGN	0.1716	0.3381	0.4122	0.5017	0.2598	0.1689	0.3391	0.3848	0.4514	0.2422	0.0810	0.1842	0.2579	0.3095	0.1675
STGCN	0.1799	0.3425	0.4279	0.5214	0.2788	0.1716	0.3453	0.3927	0.4763	0.2504	0.0961	0.2097	0.2613	0.3245	0.1712
PLSPL	0.1917	0.3678	0.4523	0.5370	0.2806	0.1889	0.3523	0.4150	0.4880	0.2542	0.1072	0.2278	0.2995	0.3401	0.1847
STAN	0.2231	0.4582	0.5734	0.6328	0.3253	0.1963	0.3798	0.4464	0.5119	0.2852	0.1104	0.2348	0.3018	0.3502	0.1869
Ours	**0.2435**	**0.5089**	**0.6143**	**0.6880**	**0.3621**	**0.2254**	**0.4417**	**0.5287**	**0.5829**	**0.3262**	**0.1357**	**0.2852**	**0.3590**	**0.4241**	**0.2103**

市和东京市）。而加利福尼亚数据集中的兴趣点分布在加利福尼亚州和内华达州，因而数据集更加稀疏。

在所有数据集上，我们的模型都大幅优于基线模型。例如，在纽约市数据集上，我们的模型 Top-1 准确率达到 24.35%，而表现最好的基线模型 STAN 为 22.31%。在 Top-5 准确率方面，我们的模型比基线模型提升约 11%；在 Top-20 准确率方面，性能提升了 8.7%。在东京也呈现出类似的结果。此外，最先进的基于注意力机制或基于 LSTM的推荐模型，如 STAN、PLSPL 和 STGCN，显著优于传统的基于马尔可夫链或基于矩阵分解的模型，如 FPMC 和个性化排序度量嵌入（PRME）。例如，STAN 在纽约市的 Top-1 准确率是 FPMC 的两倍，分别为 22.31% 和 10.03%。

加利福尼亚数据集包含 9.9k 个兴趣点和 250k 条签到记录，分布在超过 400,000 km^2 的区域内；而东京数据集包含 7.8k 个兴趣点和 361k 条签到，集中在约 2,000 km^2 的区域内。由于签到数量和兴趣点空间稀疏性方面的数据稀缺问题更加严重，模型在加利福尼亚通常无法达到在纽约市和东京上的同等性能。具体而言，STAN 在纽约市的 Top-1 准确率为 22.31%，而在加利福尼亚则降至 11.04%。我们的模型也呈现出类似趋势。在加利福尼亚数据集上，我们的模型 Top-1 准确率为 13.57%，超过所有基线模型，但远低于其在纽约市数据集上的表现。

5.3 检视轨迹流图

上一节展示了所提出框架的整体性能。本节将详细阐述轨迹流图的优势。我们将说明，与其他方法相比，我们的方法能够缓解非活跃用户和短轨迹带来的冷启动问题。随后，我们移除轨迹流图，改为通过嵌入层学习兴趣点嵌入，并比较其性能。最后，我们介绍为纽约市数据集构建的图的量化特征。

5.3.1 非活跃用户和活跃用户。给定一个数据集，将其划分为训练集、验证集和测试集后，我们仅统计训练集中每个用户的轨迹数量。然后将排名前 15% 的用户标记为最活跃用户，排名后 15% 的用户标记为非活跃用户，其余用户为普通用户。接下来，我们在测试集上评估模型，并记录不同用户组的表现。

以纽约市数据集为例。在训练集中共有 1,047 名用户，其中 137 名非活跃用户的轨迹少于 13 条，144 名最活跃用户的轨迹多于 41 条，其余 766 名为普通用户。一般而言，非活跃用户缺乏丰富的历史信息，因此很难从数据中学习。为缓解这一挑战，我们构建了全局轨迹流动图，并尝试利用其他用户的序列来改善非活跃用户的推荐结果。

表3展示了我们的模型和基线在不同用户组中的表现。与基线模型相比，我们的模型在三个用户组中的准确率均更高。对于非活跃用户，我们的模型的 Top-1 准确率和 Top-10准确率分别为 12.24% 和 43.94%，而 STGN 的对应准确率为 9.06% 和 31.24%。Top-5准确率和 Top-20 准确率也有显著提升。

表3：纽约市的冷启动（由非活跃用户导致）表现

用户组	模型	Acc@1	Acc@5	Acc@10	Acc@20
非活跃	STGN	0.0926	0.2435	0.3124	0.3763
普通	STGN	0.1778	0.3576	0.4189	0.5125
非常活跃	STGN	0.1813	0.3832	0.4274	0.5398
非活跃	Ours	0.1224	0.3471	0.4394	0.4529
普通	Ours	0.2421	0.4739	0.5422	0.6430
非常活跃	Ours	0.2692	0.5639	0.6995	0.7762

5.3.2 短轨迹和长轨迹。短轨迹给下一兴趣点推荐带来了另一项挑战。短轨迹的信息匮乏体现在其时空上下文有限，尤其是轨迹中的已知签到数量仅为一两个时。我们根据数据集动态定义短轨迹。具体而言，我们按长度对测试集中的轨迹排序，并将前15%标记为长轨迹，将后15%标记为短轨迹。

表4展示了我们的模型和STGN在纽约市数据集的三个轨迹组上的表现。我们的模型在短轨迹和长轨迹上的推荐性能没有显著差异。例如，短轨迹和长轨迹的Top-1准确率分别为21.86%和24.52%，而短轨迹的Top-20准确率为57.82%。然而，作为一种扩展的长短期记忆网络模型，STGN对轨迹长度相当敏感，在短轨迹上的表现不尽如人意，Top-1准确率为7.23%，Top-20准确率为33.28%。

表4：纽约市冷启动（由短轨迹导致）性能

轨迹	模型	Acc@1	Acc@5	Acc@10	Acc@20
短轨迹	STGN	0.0723	0.2117	0.2784	0.3328
中等长度轨迹	STGN	0.1921	0.3471	0.4397	0.5231
长轨迹	STGN	0.1934	0.3516	0.4380	0.5322
短轨迹	Ours	0.2186	0.4561	0.5269	0.5782
中等长度轨迹	Ours	0.2441	0.4927	0.5881	0.6519
长轨迹	Ours	0.2452	0.5378	0.6698	0.7681

5.3.3 移除轨迹流图。作为对比，我们从模型中移除轨迹流图，并通过嵌入层学习兴趣点表示，以展示全局轨迹流图的作用。表5展示了在三类用户组和三类轨迹组上，不使用轨迹流图的模型表现。综合表3、表4和表5可以发现，移除轨迹流图后，非活跃用户的 Top-10 准确率从 42.11% 降至 34.70%，短轨迹的表现则从 52.06% 降至 47.21%。这些实验验证了当 k 较大时，全局轨迹流图有助于提升非活跃用户和短轨迹的 Top-k 准确率。然而，对于冷启动问题，轨迹流图对 Top-1 准确率的提升并不明显。这一现象符合我们的假设：当 k 足够大时，融入通用移动模式最有助于提升 Top-k 准确率。因此，即便 Top-1 准确率差异不大，Top-10 表现确实有所提升。

表5：所提出的不含轨迹流图的模型在纽约市上的性能

	模型	Acc@1	Acc@5	Acc@10
非活跃用户	我们的无图版本	0.1105	0.2983	0.3470
普通用户	我们的无图版本	0.2094	0.4228	0.5010
非常活跃用户	我们的无图版本	0.2488	0.5271	0.6815
短轨迹	我们的无图版本	0.1914	0.4059	0.4721
中等长度轨迹	我们的无图版本	0.2129	0.4490	0.5232
长轨迹	我们的无图版本	0.2012	0.4882	0.6186

5.3.4 轨迹流图分析。我们以纽约市数据集为案例，分析所构建的轨迹流图 G 的结构属性。该图包含 4980 个节点和

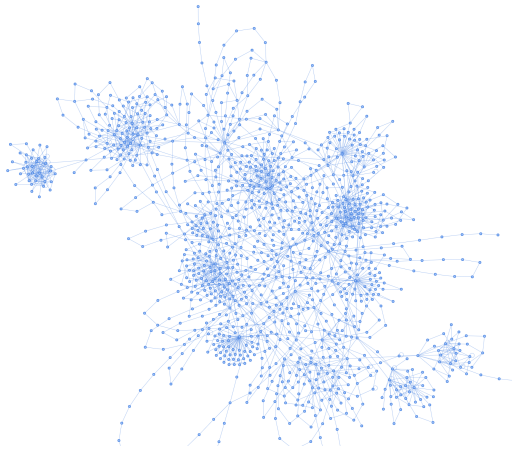


图 4：纽约市的部分轨迹流图（为便于可视化，已移除方向和边权重）

37,756 条边。平均入度和出度均为 7.58。从宏观角度来看，下一步通常约有 7.58 种可能的选择。平均边权重为 1.91。也就是说，如果兴趣点 A 和兴趣点 B 出现在连续两次签到中，那么这种签到顺序平均出现 1.91 次。此外，该图的平均聚类系数为 0.1866，表明其具有显著的聚类特性。图 4 展示了纽约市数据集的部分图结构（未显示边权重和方向）。图示表明，该图包含值得进一步探索的非平凡结构信息。具体而言，图中包含若干清晰可见的社区，但也存在许多长路径，这表明该图偏离了典型的小世界网络。

5.4 消融研究

我们进行消融研究，以评估各个提出的组件对模型在纽约市数据集上最终性能的影响。具体而言，我们进行了七项实验：1）完整模型；2）移除轨迹流图，并通过嵌入层学习兴趣点嵌入；3）移除 Transformer 编码器，改用长短期记忆网络处理序列数据；4）移除时间和类别信息；5）移除图卷积网络，并采用随机游走从转移图中学习节点嵌入；6）不融合用户嵌入和兴趣点嵌入，而是直接将它们拼接；7）移除时间解码器和类别解码器。在每项实验中，其余设置均与完整模型保持一致，且只替换或移除一个组件。结果如表 6 所示。

完整模型的表现最佳。对于其余组件，结果表明，轨迹流图和图卷积网络对整体表现的影响大于其他组件。例如，不使用轨迹流图，或使用随机游走来学习兴趣点嵌入时，Top-1准确率会从24.35%降至21.63%和20.62%。其他组件也有助于最终的推荐结果。例如，Transformer可以通过自注意力模块自动利用历史数据中的有用信息，同时所需训练参数更少，表现也更好。

表 6：消融研究：比较完整模型与 6 个变体

	Acc@1	Acc@5	Acc@10 Acc@20 平均倒数排名		
完整模型	**0.2435**	**0.5089**	**0.6143**	**0.6880**	**0.3621**
无图	0.2163	0.4512	0.5248	0.6402	0.3398
无 Transformer	0.2221	0.4617	0.5520	0.6261	0.3317
无时间与类别	0.2296	0.4804	0.5489	0.6484	0.3495
无图卷积网络	0.2062	0.4582	0.5601	0.6589	0.3187
无融合	0.2355	0.4731	0.5726	0.6614	0.3533
单解码器	0.2207	0.4721	0.5722	0.6591	0.3443

6 结论

本文提出了GETNext，这是首个利用全局图结构进行基于图的学习的下一兴趣点推荐模型。具体而言，我们引入了轨迹流图，以捕捉用户的通用移动模式，从而解决非活跃用户和短轨迹问题。我们设计了精细的嵌入来编码时空上下文，包括用户-兴趣点嵌入以及时间-类别嵌入。我们将所有嵌入输入Transformer模型，并通过转移注意力图进一步增强其输出。通过在三个真实世界数据集上进行一系列实验，我们证明该模型的表现显著优于所有当前最先进模型，并验证了模型各个组件的优势。未来，我们将进一步探索并区分时间模式，例如工作日与周末之间的差异。另一项未来工作是根据用户行为对用户进行分类，并为每类用户构建轨迹流图。

致谢。本研究得到政府资助的马斯登基金委员会（MFP-UOA2123）支持，并由新西兰皇家学会管理。

参考文献

[1] Buru Chang、Gwanghoon Jang、Seoyoon Kim 和 Jaewoo Kang。2020年。学习用于兴趣点推荐的基于图的地理潜在表示。载于第*29*届 *ACM* 信息与知识管理国际会议论文集。135–144。[2] Chen Cheng、Haiqin Yang、Michael R Lyu 和 Irwin King。2013年。你接下来想去哪里：连续兴趣点推荐。载于第二十三届国际人工智能联合会议。[3] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、Xiaohua Zhai、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly 等。2020年。一张图像胜过16×16 个词：用于大规模图像识别的 Transformer。*arXiv* 预印本 *arXiv:2010.11929*（2020年）。[4] Jie Feng、Yong Li、Chao Zhang、Funing Sun、Fanchao Meng、Ang Guo 和 Depeng Jin。2018年。DeepMove：利用注意力循环网络预测人类移动性。载于*2018*年万维网大会论文集。1459–1468。[5] Shanshan Feng、Xutao Li、Yifeng Zeng、Gao Cong、Yeow Meng Chee 和 Quan Yuan。2015年。用于下一新兴趣点推荐的个性化排序度量嵌入。载于第二十四届国际人工智能联合会议。[6] Shanshan Feng、Lucas Vinh Tran、Gao Cong、Lisi Chen、Jing Li 和 Fan Li。2020年。HME：一种用于下一兴趣点推荐的双曲度量嵌入方法。载于第*43*届 *ACM SIGIR* 信息检索研究与发展会议论文集。1429–1438。[7] Sepp Hochreiter 和 Jürgen Schmidhuber。1997年。长短期记忆。《神经计算》9，8（1997年），1735–1780。[8] Shuhui Jiang、Xueming Qian、Jialie Shen、Yun Fu 和 Tao Mei。2015 年。基于作者主题模型的协同过滤，用于个性化兴趣点推荐。《*IEEE*多媒体汇刊》17，6（2015年），907–918。[9] Seyed Mehran Kazemi、Rishab Goel、Sepehr Eghbali、Janahan Ramanan、Jaspreet Sahota、Sanjay Thakur、Stella Wu、Cathal Smyth、Pascal Poupart 和 Marcus Brubaker。2019年。Time2Vec：学习时间的向量表示。*arXiv* 预印本 arXiv:1907.05321（2019年）。[10] Thomas N Kipf 和 Max Welling。2017年。利用图卷积网络进行半监督分类。（2017年）。[11] Yehuda Koren、Robert Bell 和 Chris Volinsky。2009年。推荐系统的矩阵分解技术。《Computer》42，8（2009年），30–37。[12] Yang Li、Tong Chen、Hongzhi Yin 和 Zi Huang。2021年。通过迭代式 Seq2Graph 增强发现用于下一兴趣点推荐的协同信号。载于国际人工智能联合会议。[13] Bryan Lim、Sercan Ö Arık、Nicolas Loeff 和 Tomas Pfister。2021年。用于可解释的多时域时间序列预测的时间融合变换器。《国际预测学杂志》（2021年）。[14] Qiang Liu、Shu Wu、Liang Wang 和 Tieniu Tan。2016年。预测下一个位置：结合时空上下文的循环模型。载于第三十届 AAAI 人工智能大会。[15] Yanchi Liu、Chuanren Liu、Bin Liu、Meng Qu 和 Hui Xiong。2016年。通过时间间隔评估实现统一的兴趣点推荐。载于第22届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。1015–1024。[16] Yingtao Luo、Qiang Liu 和 Zhaocheng Liu。2021年。STAN：用于下一位置推荐的时空注意力网络。载于2021年万维网大会论文集。2177–2185。[17] Daniel Neimark、Omri Bar、Maya Zohar 和 Dotan Asselmann。2021年。视频变换器网络。arXiv 预印本 arXiv:2102.00719 （ 2021 年）。[18] Steffen Rendle 、 Christoph Freudenthaler 和 Lars Schmidt-Thieme。2010年。通过分解个性化马尔可夫链实现下一篮子推荐。载于第19届万维网国际会议论文集。811–820。[19] Ke Sun、Tieyun Qian、Tong Chen、Yile Liang、Quoc Viet Hung Nguyen 和 Hongzhi Yin。2020年。接下来去哪里：对用户长期和短期偏好进行建模，以实现兴趣点推荐。载于 AAAI 人工智能大会论文集，第34卷。214–221。[20] Jian Tang、Meng Qu、Mingzhe Wang、Ming Zhang、Jun Yan 和 Qiaozhu Mei。2015年。LINE：大规模信息网络嵌入。载于第24届万维网国际会议论文集。1067–1077。[21] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。Attention Is All You Need。载于《神经信息处理系统进展》。5998–6008。[22] Yuxia Wu、Ke Li、Guoshuai Zhao 和 Xueming Qian。2019年。用于下一兴趣点推荐的长期和短期偏好学习。载于第28届 ACM 信息与知识管理国际会议论文集。2301–2304。[23] Yuxia Wu、Ke Li、Guoshuai Zhao 和 QIAN Xueming。2020年。用于下一兴趣点推荐的个性化长期和短期偏好学习。《IEEE 知识与数据工程汇刊》（2020年）。[24] Min Xie、Hongzhi Yin、Hao Wang、Fanjiang Xu、Weitong Chen 和 Sen Wang。2016年。学习用于基于位置的推荐的基于图的兴趣点嵌入。载于第25届 ACM 信息与知识管理国际会议论文集。15–24。[25] Dingqi Yang、Daqing Zhang、Vincent W Zheng 和 Zhiyong Yu。2014年。通过利用基于位置的社交网络中的用户时空特征，对用户活动偏好进行建模。《IEEE 系统、人类与控制论汇刊：系统》45，1（2014年），129–142。[26] Jihang Ye、Zhe Zhu 和 Hong Cheng。2013年。你的下一步行动是什么：基于位置的社交网络中的用户活动预测。载于2013年 SIAM 国际数据挖掘会议论文集。SIAM，171–179。[27] Quan Yuan、Gao Cong、Zongyang Ma、Aixin Sun 和 Nadia Magnenat Thalmann。2013年。时间感知兴趣点推荐。载于第36届 ACM SIGIR 信息检索研究与发展国际会议论文集。363–372。[28] Quan Yuan、Gao Cong 和 Aixin Sun。2014年。融合地理与时间影响的基于图的兴趣点推荐。载于第23届 ACM 信息与知识管理国际会议论文集。659–668。[29] Jia-Dong Zhang、Chi-Yin Chow 和 Yanhua Li。2014年。LORE：利用序列影响进行位置推荐。载于第22届 ACM SIGSPATIAL 地理信息系统进展国际会议论文集。103–112。[30] Kangzhi Zhao、Yong Zhang、Hongzhi Yin、Jin Wang、Kai Zheng、Xiaofang Zhou 和 Chunxiao Xing。2020年。发现用于下一兴趣点推荐的子序列模式。载于国际人工智能联合会议。3216–3222。[31] Pengpeng Zhao、Anjing Luo、Yanchi Liu、Fuzhen Zhuang、Jiajie Xu、Zhixu Li、Victor S Sheng 和 Xiaofang Zhou。2020年。接下来去哪里：用于下一兴趣点推荐的时空门控网络。《IEEE 知识与数据工程汇刊》（2020年）。[32] Shenglin Zhao、Tong Zhao、Haiqin Yang、Michael R Lyu 和 Irwin King。2016年。STELLAR：用于连续兴趣点推荐的时空潜在排序。载于第三十届 AAAI 人工智能大会。