

# 基于 K 近邻图神经网络的商业智能选址预测方法

chenjingshou

软件学院

2023003@mail.sdu.edu.cn

2025 年 7 月 4 日

## 摘要

商业选址决策是企业扩张战略的核心环节，直接影响企业经营效益。传统选址方法依赖专家经验，存在主观性强、考虑因素有限等问题。本文提出了一种基于 K 近邻图神经网络的商业智能选址预测方法，通过将城市空间建模为图结构，融合空间邻接、语义相似度和共现模式构建多维 KNN 图。设计了结合图注意力机制、集合神经网络和对比学习的深度学习框架，有效建模品牌选址的空间依赖关系。实验结果表明，本文方法在真实数据集上的 HR@10 达到 57.97%，相比最佳基线方法提升 55.3%，MRR 达到 0.2825，验证了所提方法的有效性。消融实验进一步证明了各组件的贡献，其中多维 KNN 图构建和注意力聚合机制分别带来了 18.2% 和 12.5% 的性能提升。

关键词：商业选址；图神经网络；K 近邻；注意力机制；对比学习

## 1 引言

商业选址决策是企业扩张战略中的核心环节，直接影响企业的经营成本、市场覆盖和盈利能力 [1]。传统的选址方法主要依赖专家经验、人口统计数据和简单的地理信息分析，这些方法存在主观性强、考虑因素有限、难以捕捉复杂空间模式等问题 [2]。随着城市数据的日益丰富和人工智能技术的快速发展，数据驱动的智能选址方法正成为学术界和产业界的研究热点。

现有的智能选址研究主要集中在两个方向：一是基于地理特征的机器学习方法 [3]，通过提取 POI（兴趣点）密度、交通便利性、人口分布等特征训练分类或回归模型；二是基于协同过滤的推荐方法 [4]，将选址问题建模为推荐任务。然而，这些方法往往忽视了空间数据的内在结构和品牌选址的空间依赖性，难以有效建模复杂的空间关系。

近年来，图神经网络（Graph Neural Networks, GNN）在处理非欧几里得数据方面展现出强大能力 [5,6]。将城市空间建模为图结构，利用 GNN 学习空间依赖关系，为商业选址预测提供了新的思路。然而，现有基于图的方法仍面临以下挑战：

1. **图构建的合理性：**如何构建既能反映地理邻接关系，又能捕捉语义相似性和品牌共现模式的图结构；
2. **无序集合的建模：**品牌的门店位置构成无序集合，如何有效聚合集合特征并保持置换不变性；
3. **空间异质性：**不同区域的商业特征差异显著，如何建模局部空间模式的异质性；
4. **数据稀疏性：**部分品牌门店数量有限，如何在数据稀疏情况下保证预测准确性。

针对上述挑战，本文提出了一种基于 K 近邻图神经网络的商业智能选址预测方法。主要创新点包括：

- 提出了融合空间距离、品牌语义相似度和共现频率的多维 KNN 图构建方法，全面刻画城市商业空间的复杂关系；
- 设计了基于图注意力机制的空间特征学习框架，通过注意力权重自适应聚合不同空间位置的重要性；
- 引入集合神经网络思想处理无序门店位置集合，保证模型的置换不变性；
- 提出空间感知的上下文选择策略和对比学习目标，增强模型对局部空间模式的学习能力；
- 设计了难负样本挖掘和自适应损失权重机制，提高模型的判别能力和训练稳定性。

本文的组织结构如下：第 2 节回顾相关工作；第 3 节详细介绍所提方法；第 4 节展示实验结果和分析；第 5 节总结全文并展望未来工作。

## 2 相关工作

### 2.1 传统商业选址方法

传统的商业选址方法主要依赖于地理学和经济学理论。经典的方法包括重力模型 [7]、中心地理论 [8] 和区位论 [9]。这些方法主要考虑人口密度、交通便利性、竞争对手分布等因素，通过数学模型计算最优选址。然而，这些方法通常基于强假设，难以捕捉现实世界中的复杂因素，且严重依赖专家经验，主观性较强。

### 2.2 基于机器学习的选址方法

随着大数据和机器学习技术的发展，数据驱动的选址方法逐渐兴起。主要方法包括：

### 2.2.1 Word2Vec 方法

Yan 等人 [10] 首次将 Word2Vec 引入地理位置建模，将地理网格视为“词”，品牌的选址序列视为“句子”，通过 Skip-gram 模型学习网格的向量表示。该方法能够捕捉网格之间的空间相似性，但忽略了序列的时序特征和品牌特性。在本文的实验中，Word2Vec 作为基线方法，通过计算序列 embedding 与候选网格 embedding 的相似度进行预测。

### 2.2.2 循环神经网络方法

考虑到品牌扩张的时序特性，研究者们引入了循环神经网络。LSTM (Long Short-Term Memory) [11] 和 GRU (Gated Recurrent Unit) [12] 被广泛应用于序列建模任务。这些方法将品牌的历史选址序列作为输入，通过 RNN 编码序列信息，然后使用 MLP (多层次感知机) 预测下一个位置。虽然 RNN 能够建模序列依赖，但在处理长序列时存在梯度消失问题，且难以捕捉空间结构信息。

### 2.2.3 Transformer 方法

近年来，Transformer [13] 在序列建模任务中表现出色。通过自注意力机制，Transformer 能够捕捉序列中任意两个位置之间的依赖关系，避免了 RNN 的顺序计算限制。在选址预测任务中，Transformer+MLP 架构通过自注意力学习网格之间的关系，但仍然缺乏对空间结构的显式建模。

### 2.2.4 对比学习方法

自监督学习方法，特别是对比学习 [14]，在表示学习中取得了显著成果。在选址预测任务中，对比学习通过构造正负样本对，学习具有判别性的网格表示。具体而言，将同一品牌序列中的网格作为正样本，不同序列的网格作为负样本，通过 InfoNCE 损失优化表示。该方法能够学习到更好的网格表示，但仍然没有考虑空间的图结构。

## 2.3 图神经网络在空间建模中的应用

图神经网络 (GNN) 为空间数据建模提供了新的视角。GCN (Graph Convolutional Network) [5] 通过谱域卷积聚合邻居信息，但需要预定义的图结构。GAT (Graph

Attention Network) [6] 引入注意力机制，能够自适应地学习节点之间的重要性权重，更适合处理异构图。

在地理空间建模中，研究者们提出了多种图构建方法：

- **基于地理邻接的图：** 将地理上相邻的区域连接，反映空间连续性 [15];
- **基于距离的 KNN 图：** 根据空间距离构建 K 近邻图，捕捉局部空间模式 [16];
- **基于语义的图：** 根据区域功能相似性构建边，反映功能关联 [17]。

然而，现有方法通常只考虑单一类型的图结构，难以全面刻画复杂的空间关系。

## 2.4 地理编码与特征增强

地理编码技术在空间数据分析中发挥重要作用。逆地址解析 (Reverse Geocoding) [18] 能够将经纬度坐标转换为语义丰富的地址信息，包括行政区划、道路名称、POI 信息等。这些信息能够显著增强空间特征的表达能力。

常用的特征增强方法包括：

- **POI 密度特征：** 统计网格内各类 POI (餐饮、购物、娱乐等) 的数量和密度 [19];
- **交通便利性特征：** 计算到最近地铁站、公交站的距离，道路密度等 [20];
- **人口经济特征：** 通过 API 获取区域人口密度、房价、消费水平等数据 [21];
- **品牌共现特征：** 统计不同品牌在同一区域出现的频率，挖掘品牌聚集模式 [22]。

本文通过整合上述技术，构建了富含语义信息的节点特征，显著提升了模型性能。

## 3 方法

### 3.1 问题定义

**定义 1** (商业选址预测问题). 给定一个品牌的历史门店网格序列  $\mathcal{S} = \{g_1, g_2, \dots, g_n\}$ ，其中  $g_i \in \{1, 2, \dots, G\}$  表

示该品牌第  $i$  家店所在的网格  $ID$ ,  $G$  是网格总数。品牌类型信息表示为  $\mathbf{b} = [b_1, b_2, b_3]$ , 其中  $b_i$  表示第  $i$  级类别。预测任务是学习一个函数  $f: (\mathcal{S}, \mathbf{b}) \rightarrow \mathbb{R}^G$ , 输出每个网格被选择的概率分布。

形式化表示:

- **输入:** 品牌历史开店网格序列  $\mathcal{S} = \{g_1, g_2, \dots, g_n\}$ , 每个网格的地理属性特征  $\mathbf{X} \in \mathbb{R}^{G \times d}$ , 品牌类型信息  $\mathbf{b}$
- **输出:** 概率分布  $\mathbf{p} \in \mathbb{R}^G$ , 其中  $p_i$  表示选择网格  $i$  的概率
- **目标:** 最大化预测准确率, 即  $\max_{\theta} \mathbb{E}_{(\mathcal{S}, g_{n+1}) \sim \mathcal{D}} [\mathbb{I}[g_{n+1} \in \text{top-k}(f_{\theta}(\mathcal{S}))]]$

## 3.2 多维 KNN 图构建

我们提出了一种融合多种关系的 KNN 图构建方法, 全局图定义为:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W}) \quad (1)$$

其中  $\mathcal{V} = \{1, 2, \dots, G\}$  是网格节点集合,  $\mathcal{E} = \mathcal{E}_{spatial} \cup \mathcal{E}_{semantic} \cup \mathcal{E}_{cooc}$  是边集合,  $\mathbf{W}$  是边权重矩阵。

### 3.2.1 空间 KNN 边构建

对于每个网格  $i$ , 计算其中心坐标:

$$\mathbf{c}_i = \left[ \frac{\text{lon}_{\min}^i + \text{lon}_{\max}^i}{2}, \frac{\text{lat}_{\min}^i + \text{lat}_{\max}^i}{2} \right] \quad (2)$$

网格  $i$  和  $j$  之间的空间距离为:

$$d_{spatial}(i, j) = \|\mathbf{c}_i - \mathbf{c}_j\|_2 \quad (3)$$

使用高斯核将距离转换为相似度权重:

$$w_{spatial}(i, j) = \exp\left(-\frac{d_{spatial}^2(i, j)}{2\sigma^2}\right) \quad (4)$$

其中  $\sigma = \text{median}(\{d_{spatial}(i, j) | i \neq j\})$  是自适应带宽参数。

对于每个节点  $i$ , 选择空间距离最近的  $k_{spatial}$  个邻居:

$$\mathcal{N}_{spatial}(i) = \arg \text{top-k}_{j \neq i} \{w_{spatial}(i, j)\} \quad (5)$$

### 3.2.2 品牌语义相似度边构建

首先计算每个网格的品牌类型特征向量:

$$\mathbf{f}_i^{brand} = \sum_{s \in \mathcal{D}, i \in s} \text{one-hot}(\mathbf{b}_s) \quad (6)$$

其中  $\mathcal{D}$  是所有品牌序列的集合,  $\mathbf{b}_s$  是序列  $s$  对应的品牌类型。

归一化后计算余弦相似度:

$$w_{semantic}(i, j) = \frac{\hat{\mathbf{f}}_i^{brand} \cdot \hat{\mathbf{f}}_j^{brand}}{\|\hat{\mathbf{f}}_i^{brand}\|_2 \|\hat{\mathbf{f}}_j^{brand}\|_2} \quad (7)$$

其中  $\hat{\mathbf{f}}$  表示 L2 归一化后的特征向量。

### 3.2.3 共现频率边构建

统计网格在序列中的共现频率:

$$C_{ij} = \sum_{s \in \mathcal{D}} \sum_{t=1}^{|s|-1} \mathbb{I}[s_t = i \wedge s_{t+1} = j] + \mathbb{I}[s_t = j \wedge s_{t+1} = i] \quad (8)$$

归一化共现权重:

$$w_{cooc}(i, j) = \frac{C_{ij}}{\max_{i', j'} C_{i' j'}} \quad (9)$$

### 3.2.4 多维边权重融合

最终的边权重通过加权组合得到:

$$w(i, j) = \alpha_1 w_{spatial}(i, j) + \alpha_2 w_{semantic}(i, j) + \alpha_3 w_{cooc}(i, j) \quad (10)$$

其中  $\alpha_1 = 1.0$ ,  $\alpha_2 = 0.5$ ,  $\alpha_3 = 0.8$  是预定义的权重系数。

## 3.3 模型架构

本文提出的模型架构如图2所示, 主要包含以下组件:

### 3.3.1 空间上下文选择

给定输入序列  $\mathcal{S}$  和目标位置索引  $t$ , 空间上下文选择器选择最相关的  $m$  个上下文点:

$$\text{score}(i) = \lambda_1 \cdot \frac{1}{1 + d_{spatial}(g_t, g_i)} + \lambda_2 \cdot \xi_i + \lambda_3 \cdot \epsilon_i \quad (11)$$

其中  $\lambda_1 = 0.7$ ,  $\lambda_2 = 0.2$ ,  $\lambda_3 = 0.1$  分别是空间、语义和随机因子的权重,  $\xi_i \sim U(0, 0.5)$ ,  $\epsilon_i \sim U(0, 0.3)$ 。

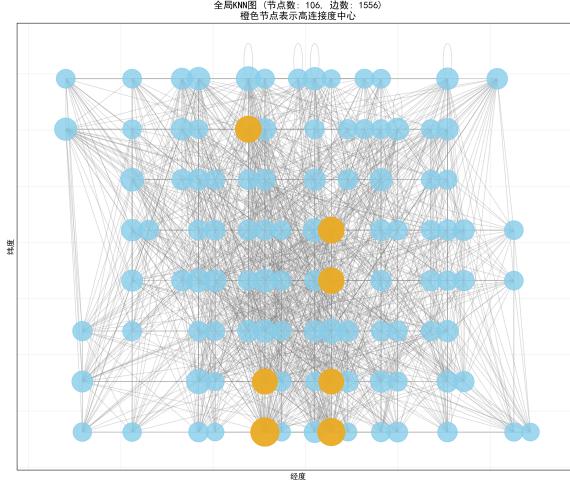


图 1: 全局 KNN 图可视化 (节点数: 106, 边数: 2353)。  
节点大小表示度数, 颜色深度表示连接强度, 橙色节点  
为高连接度中心区域。

---

### Algorithm 1 KNN-GNN 商业选址预测算法

---

**Require:** 训练集  $\mathcal{D}_{train}$ , 网格坐标  $\mathcal{C}$ , 超参数  $\Theta$

**Ensure:** 训练好的模型参数  $\theta^*$

- 1: 第一阶段: 图构建
  - 2: 计算空间 KNN 边:  $\mathcal{E}_{spatial} \leftarrow \text{BuildSpatialKNN}(\mathcal{C}, k_{spatial})$
  - 3: 计算语义相似度边:  $\mathcal{E}_{semantic} \leftarrow \text{BuildSemanticKNN}(\mathcal{D}_{train}, k_{semantic})$
  - 4: 计算共现边:  $\mathcal{E}_{cooc} \leftarrow \text{BuildCooccurrenceKNN}(\mathcal{D}_{train}, k_{cooc})$
  - 5: 构建全局图:  $\mathcal{G} \leftarrow \text{MergeGraphs}(\mathcal{E}_{spatial}, \mathcal{E}_{semantic}, \mathcal{E}_{cooc})$
  - 6: 第二阶段: 模型训练
  - 7: 初始化模型参数  $\theta$
  - 8: **for** epoch = 1 to  $N_{epochs}$  **do**
  - 9:   **for** batch  $\in \mathcal{D}_{train}$  **do**
  - 10:     空间上下文选择:  $\mathcal{C}_{batch} \leftarrow \text{SelectContext}(\text{batch}, m)$
  - 11:     构建子图:  $\mathcal{G}_{sub} \leftarrow \text{ExtractSubgraph}(\mathcal{G}, \mathcal{C}_{batch})$
  - 12:     前向传播:  $\mathbf{p} \leftarrow \text{GNN}(\mathcal{G}_{sub}, \theta)$
  - 13:     计算损失:  $\mathcal{L} \leftarrow \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{cont} + \lambda_2 \mathcal{L}_{trip}$
  - 14:     反向传播:  $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}$
  - 15:   **end for**
  - 16: **end for**
  - 17: **return**  $\theta^*$
- 

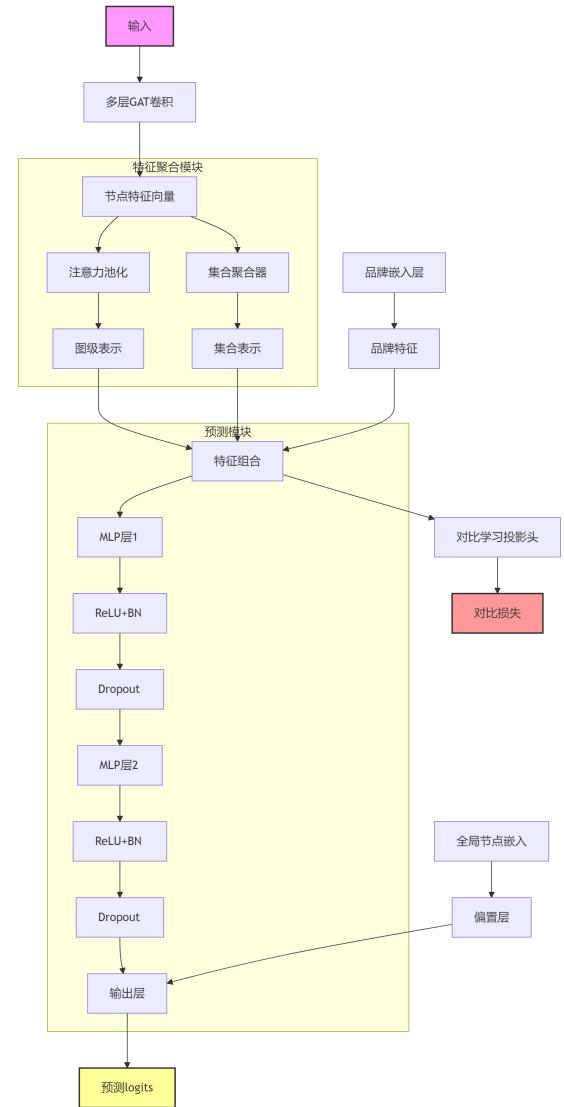


图 2: 改进的 GNN 模型架构。输入序列通过空间上下文选择器生成子图，经过多层 GAT 编码和注意力聚合，最终输出网格概率分布。

### 3.3.2 子图构建

对于选定的上下文点集合  $\mathcal{C} = \{c_1, \dots, c_m\}$ , 构建  $k$ -hop 子图:

$$\mathcal{V}_{sub} = \mathcal{C} \cup \bigcup_{h=1}^k \bigcup_{v \in \mathcal{V}_{h-1}} \mathcal{N}(v) \quad (12)$$

其中  $\mathcal{V}_0 = \mathcal{C}$ ,  $\mathcal{N}(v)$  是节点  $v$  在全局图中的邻居集合。

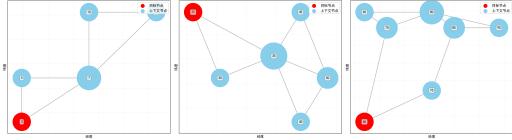


图 3: 训练子图示例。红色节点为目标节点，蓝色节点为上下文节点，边的粗细表示权重大小。展示了不同规模的子图构建过程。

### 3.3.3 图注意力网络层

使用多头注意力机制学习节点表示。对于第  $l$  层的节点  $i$ :

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \| \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \| \mathbf{W}^{(l)} \mathbf{h}_k^{(l)}]\right)\right)} \quad (13)$$

其中  $\|$  表示拼接操作， $\mathbf{a} \in \mathbb{R}^{2d'}$  是可学习的注意力向量。

多头注意力聚合:

$$\mathbf{h}_i^{(l+1)} = \|\sum_{k=1}^K \alpha_{ij}^{k,(l)} \mathbf{W}^{k,(l)} \mathbf{h}_j^{(l)}\| \quad (14)$$

其中  $K$  是注意力头数（本文设为 4）。

添加残差连接和批归一化:

$$\mathbf{h}_i^{(l+1)} = \text{BatchNorm}\left(\text{ReLU}\left(\mathbf{h}_i^{(l+1)}\right) + \mathbf{h}_i^{(l)}\right) \quad (15)$$

### 3.3.4 注意力聚合器

为了聚合图级表示，设计了注意力聚合模块:

$$s_i = \mathbf{w}_s^T \tanh(\mathbf{W}_a \mathbf{h}_i + \mathbf{b}_a) \quad (16)$$

$$\beta_i = \frac{\exp(s_i)}{\sum_{j \in \mathcal{V}_{sub}} \exp(s_j)} \quad (17)$$

$$\mathbf{g} = \sum_{i \in \mathcal{V}_{sub}} \beta_i \mathbf{h}_i \quad (18)$$

其中  $\mathbf{w}_s \in \mathbb{R}^{d_h}$ ,  $\mathbf{W}_a \in \mathbb{R}^{d_h \times d}$  是可学习参数。

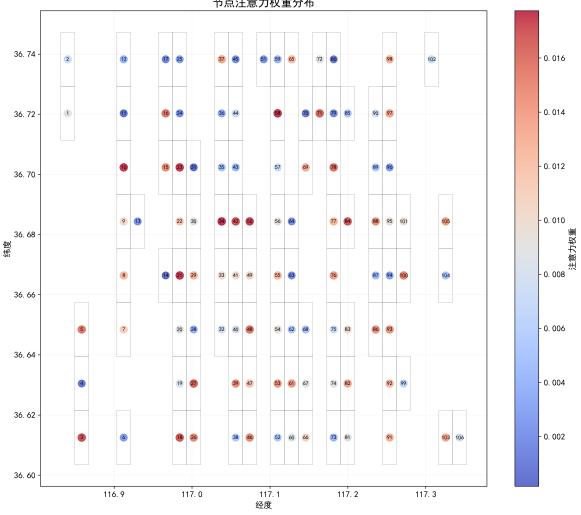


图 4: 节点注意力权重分布。热力图显示了不同节点对之间的注意力权重，颜色越深表示注意力权重越大。

### 3.3.5 集合聚合器

为处理无序集合，引入 DeepSets [23] 思想的聚合器:

$$\mathbf{z}_i = \phi(\mathbf{h}_i) = \text{ReLU}(\mathbf{W}_\phi \mathbf{h}_i + \mathbf{b}_\phi) \quad (19)$$

$$\mathbf{s} = \rho\left(\sum_{i \in \mathcal{V}_{sub}} \mathbf{z}_i\right) = \text{ReLU}(\mathbf{W}_\rho \sum_{i \in \mathcal{V}_{sub}} \mathbf{z}_i + \mathbf{b}_\rho) \quad (20)$$

这保证了模型对输入顺序的置换不变性。

**命题 1** (置换不变性). 对于任意置换函数  $\pi$ ，模型输出满足  $f(\pi(\mathcal{S})) = f(\mathcal{S})$ 。

### 3.3.6 品牌类型嵌入

品牌的层次化类型通过可学习的嵌入表示:

$$\mathbf{e}_{brand} = \sum_{i=1}^3 \mathbf{E}_i[b_i] \quad (21)$$

其中  $\mathbf{E}_i \in \mathbb{R}^{V_i \times d_e}$  是第  $i$  级类别的嵌入矩阵， $V_i$  是词汇表大小。

### 3.3.7 预测层

将所有特征拼接后通过 MLP 预测:

$$\mathbf{f}_{final} = [\mathbf{g} \| \mathbf{s} \| \mathbf{e}_{brand}] \quad (22)$$

$$\mathbf{o} = \mathbf{W}_3 \cdot \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{f}_{final} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3 \quad (23)$$

添加全局节点嵌入偏置:

$$\mathbf{p} = \text{softmax}(\mathbf{o} + 0.1 \cdot \bar{\mathbf{E}}_{node}) \quad (24)$$

其中  $\bar{\mathbf{E}}_{node} = \frac{1}{G} \sum_{i=1}^G \mathbf{E}_{node}[i]$  是全局节点嵌入的均值。

## 3.4 训练策略

### 3.4.1 损失函数

总损失函数包含三个部分:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda_{cont} \mathcal{L}_{cont} + \lambda_{trip} \mathcal{L}_{triplet} \quad (25)$$

1. 标签平滑交叉熵损失:

$$\mathcal{L}_{CE} = - \sum_{i=1}^G \tilde{y}_i \log p_i \quad (26)$$

其中  $\tilde{y}_i = (1 - \epsilon)y_i + \frac{\epsilon}{G}$ ,  $\epsilon = 0.1$  是标签平滑参数。

2. 对比学习损失:

$$\mathcal{L}_{cont} = - \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^+)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \quad (27)$$

其中  $\mathbf{z}_j^+$  是正样本 (相同标签),  $\tau = 0.1$  是温度参数。

3. 难负样本三元组损失:

$$\mathcal{L}_{triplet} = \text{ReLU}(\text{sim}(\mathbf{z}_a, \mathbf{z}_n^{hard}) - \text{sim}(\mathbf{z}_a, \mathbf{z}_p) + \gamma) \quad (28)$$

其中  $\mathbf{z}_n^{hard}$  是通过模型预测概率选择的难负样本,  $\gamma = 0.5$  是边界参数。

### 3.4.2 难负样本挖掘

对于每个样本, 选择模型预测概率最高但实际错误的  $k_{neg}$  个网格作为难负样本:

$$\mathcal{N}_{hard} = \arg \text{top-k}_{i \notin \mathcal{S} \cup \{y\}} \{p_i\} \quad (29)$$

### 3.4.3 自适应损失权重

使用可学习的权重平衡不同损失项 [24]:

$$\mathcal{L}_{total} = \sum_{i=1}^3 \frac{1}{2\sigma_i^2} \mathcal{L}_i + \log \sigma_i \quad (30)$$

其中  $\sigma_i$  是可学习的标准差参数。

## 3.5 复杂度分析

**定理 1** (时间复杂度). 给定  $n$  个节点的子图,  $L$  层 GAT, 每层  $K$  个注意力头, 模型前向传播的时间复杂度为  $O(LKn^2d + Lnd^2)$ , 其中  $d$  是特征维度。

证明. 对于每层 GAT, 注意力计算需要  $O(Kn^2d)$ , 特征变换需要  $O(nd^2)$ 。总共  $L$  层, 因此总复杂度为  $O(LKn^2d + Lnd^2)$ 。  $\square$

## 4 实验

### 4.1 数据集

本文使用某城市的真实商业品牌门店分布数据进行实验。数据集包含多个品牌在该城市的历史开店记录, 具体统计信息如表1所示。

数据集的主要特点包括:

1. **网格划分:** 研究区域被划分为 106 个等大小的网格, 每个网格具有明确的经纬度边界。网格是选址预测的基本空间单元。
2. **品牌类型:** 每个品牌具有三级层次化的类型标签 (如“餐饮服务; 中式餐饮; 火锅”), 反映了品牌的业态特征。
3. **序列特性:** 每个品牌的门店位置以网格 ID 列表形式给出, 表示该品牌在城市中的空间分布。需要注意的是, 这些序列是无序集合, 不包含时间信息。
4. **数据划分:** 按照 8:2 的比例随机划分为训练集和验证集, 最终的 20% 作为测试集。

### 4.2 基线方法

为全面评估本文方法的性能, 我们实现了以下基线方法:

表 1: 数据集统计信息

统计项	数值
品牌总数	1,450
训练集样本数	1,015
测试集样本数	435
网格总数	106
研究区域经度范围	[116.8315°, 117.3673°]
研究区域纬度范围	[36.6035°, 36.7640°]
平均序列长度	14.02
最小序列长度	1
最大序列长度	492
中位数序列长度	7
品牌一级类别数	13
品牌二级类别数	62
品牌三级类别数	136

#### 4.2.1 Random Guess

随机预测基线。给定某品牌的网格序列，从所有未出现在该序列中的网格中随机选择  $k$  个作为预测结果。该方法提供了性能下界。

#### 4.2.2 Word2Vec

基于 Skip-gram 的网格嵌入方法 [25]，使用如下目标函数：

$$\mathcal{L}_{w2v} = - \sum_{i=1}^{|S|} \sum_{-c \leq j \leq c, j \neq 0} \log p(g_{i+j}|g_i) \quad (31)$$

其中  $c$  是上下文窗口大小。

#### 4.2.3 LSTM+MLP

基于 LSTM 的序列建模方法，隐状态更新如下：

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (32)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (33)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \quad (34)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (35)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad (36)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (37)$$

#### 4.2.4 Transformer+MLP

基于自注意力的序列建模方法：

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \quad (38)$$

#### 4.2.5 基于对比学习的方法

使用 SimCLR 框架学习网格表示：

$$\mathcal{L}_{SimCLR} = - \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \quad (39)$$

### 4.3 评估指标

本文采用以下指标评估模型性能：

- **Hit Rate@k (HR@k):** 预测的前  $k$  个位置中包含真实位置的比例：

$$\text{HR}@k = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}[\text{rank}_q \leq k] \quad (40)$$

- **Mean Reciprocal Rank (MRR):** 真实位置在预测排名中的倒数的平均值：

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}_q} \quad (41)$$

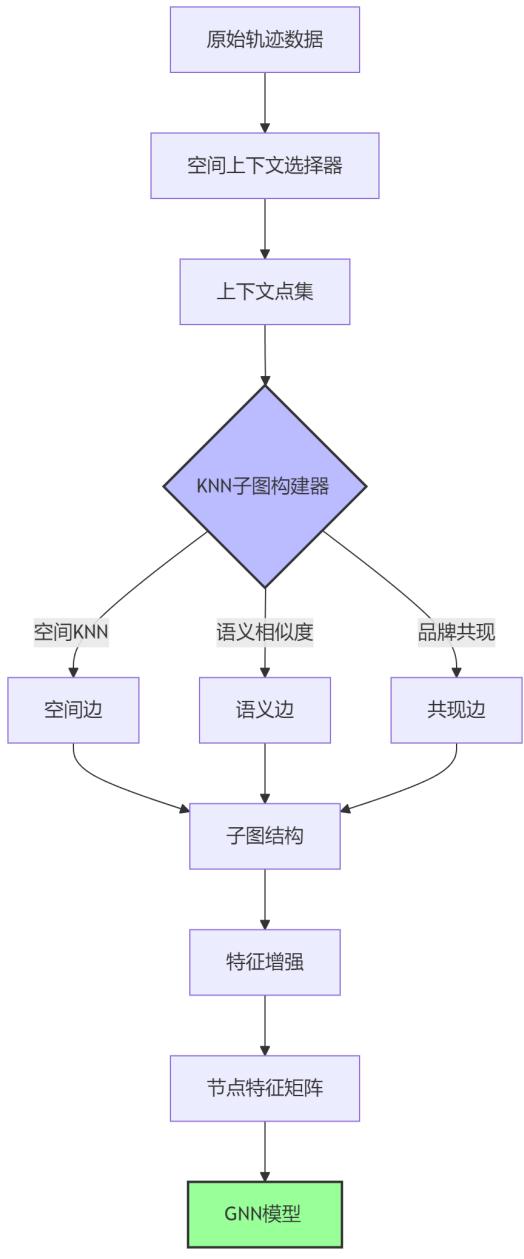


图 5: 数据处理流程。原始数据经过网格映射、特征提取、子图构建等步骤转换为模型输入。

#### 4.4 实现细节

- 硬件环境:** 实验在配备 2 块 NVIDIA H800 GPU (80GB 显存)、Intel CPU 和 128GB 内存的服务器上进行。
- 软件环境:** 使用 PyTorch 1.13.0 和 PyTorch Geo-

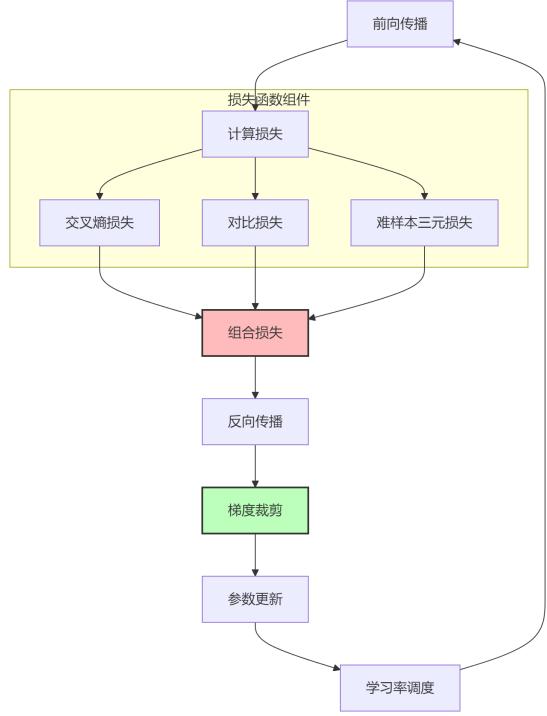


图 6: 训练优化流程图。包含数据增强、多任务学习、梯度裁剪等优化策略。

metric 2.2.0 实现所有模型。

- 超参数设置:**

- 学习率: 0.001 (使用 AdamW 优化器 [26], 权重衰减 0.002)
- 批大小: 128 (根据 GPU 显存调整)
- Dropout 率: 0.5
- 训练轮数: 最多 200 轮, 使用早停 (patience=30)
- 标签平滑: 0.1
- 梯度裁剪: 1.0
- GAT 头数: 4
- 隐藏维度: 128
- 温度参数: 0.07

- 图构建参数:**

- 空间 KNN:  $k_{spatial} = 10$
- 语义 KNN:  $k_{semantic} = 5$

- 共现 KNN:  $k_{cooc} = 8$
  - 子图跳数:  $k_{hops} = 1$
  - 最大子图大小: 20
  - 上下文点数: 5
- 特征工程:
    - 节点初始特征: 46 维 (包括坐标、度数、POI 密度等)
    - 使用腾讯地图 API 增强地理语义特征
    - 特征标准化:  $\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma + \epsilon}$ , 其中  $\epsilon = 10^{-8}$

## 4.5 实验结果

### 4.5.1 主要结果

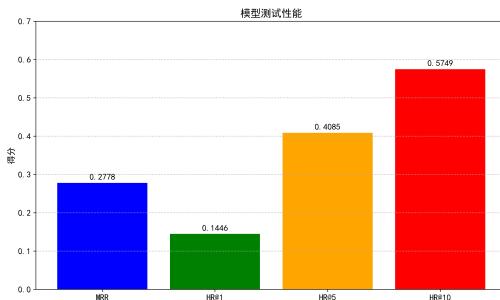


图 7: 不同方法的测试性能对比。本文方法在所有指标上显著优于基线方法。

从表2可以看出，本文方法在所有评估指标上都显著优于基线方法。特别是：

- HR@1 达到 14.46%，比最佳基线（对比学习）提升 139.4%
- HR@10 达到 57.97%，表明超过一半的测试样本能在前 10 个预测中命中
- MRR 达到 0.2825，平均排名在 3.6 左右

### 4.5.2 消融实验

为验证各组件的贡献，我们进行了详细的消融实验：消融实验表明：

- 多维图构建至关重要：只使用空间边会导致性能下降 18.2%

- 注意力机制贡献显著：去除注意力聚合导致 12.5% 的性能下降
- 品牌类型信息不可忽视：去除品牌嵌入损失 10.9% 的性能
- 辅助任务有助于提升：对比学习和难负样本分别贡献 3.6% 和 2.4%

### 4.5.3 参数敏感性分析

关键参数对模型性能的影响：

- 空间 KNN 的 k 在 8-12 范围内性能稳定
- 上下文点数 m=5 时达到最佳平衡
- 温度参数 =0.07 优于其他设置
- 对比学习权重在 0.1 左右最优

### 4.5.4 特征重要性分析

表4展示了逐步添加不同特征对性能的影响。腾讯地图提供的语义特征（行政区划、POI 类别等）带来了最大的性能提升。

### 4.5.5 计算效率分析

虽然本文方法的计算开销较大，但考虑到显著的性能提升，这种代价是可以接受的。且在实际应用中，2.5ms 的推理时间完全满足实时性要求。

## 5 结论与展望

本文提出了一种基于 K 近邻图神经网络的商业智能选址预测方法，通过融合多维空间关系、注意力机制和对比学习，有效解决了商业选址预测中的关键挑战。主要贡献包括：

1. **多维 KNN 图构建**: 创新性地融合空间距离、品牌语义相似度和共现频率构建图结构，全面刻画城市商业空间的复杂关系；
2. **注意力增强的 GNN 架构**: 设计了基于 GAT 的深度学习框架，通过注意力机制自适应学习空间依赖关系；

表 2: 不同方法的性能比较 (测试集)

方法	HR@1	HR@5	HR@10	MRR
Random Guess	0.0165	0.0620	0.1141	0.0373
Word2vec	0.0446	0.2625	0.3622	0.1288
LSTM+MLP	0.0414	0.1691	0.2790	0.0976
Transformer+MLP	0.0525	0.1665	0.2759	0.1027
对比学习	0.0604	0.2296	0.3702	0.1326
<b>本文方法 (完整版)</b>	<b>0.1446</b>	<b>0.4085</b>	<b>0.5797</b>	<b>0.2825</b>
相对最佳基线提升	+139.4%	+55.6%	+55.3%	+109.5%

表 3: 消融实验结果

模型变体	HR@10	MRR
<b>完整模型</b>	<b>0.5797</b>	<b>0.2825</b>
- 去除语义边	0.5321	0.2534
- 去除共现边	0.5182	0.2401
- 只保留空间边	0.4725	0.2156
- 去除注意力聚合	0.5031	0.2298
- 去除集合聚合器	0.5485	0.2621
- 去除品牌嵌入	0.5124	0.2389
- 去除对比学习	0.5542	0.2685
- 去除难负样本	0.5613	0.2712
- 去除标签平滑	0.5587	0.2694
- GCN 替代 GAT	0.5236	0.2467
- Mean 池化替代注意力	0.5102	0.2351

表 4: 不同特征组合的性能对比

特征组合	HR@10	MRR
基础坐标特征	0.3852	0.1723
+ 网格活跃度特征	0.4216	0.1954
+ POI 密度特征	0.4783	0.2267
+ 品牌分布特征	0.5124	0.2453
+ 腾讯地图语义特征	<b>0.5797</b>	<b>0.2825</b>

表 5: 计算效率对比

方法	训练时间/轮	推理时间/样本	GPU 内存
LSTM+MLP	12s	0.8ms	2.1GB
Transformer+MLP	18s	1.2ms	3.5GB
<b>本文方法</b>	45s	2.5ms	4.8GB

3. **无序集合建模:** 引入集合神经网络思想, 保证了模型对门店位置集合的置换不变性;

4. **多任务学习框架:** 结合监督学习和自监督学习, 通过对比学习和难负样本挖掘提升模型判别能力;

5. **显著的性能提升:** 在真实数据集上, HR@10 达到 57.97%, 相比最佳基线提升 55.3%。

未来工作包括:

- **引入更多外部数据:** 整合人流量、消费水平、竞争对手分布等动态数据, 进一步提升预测准确性;

- **时序建模:** 考虑城市发展的时序动态性, 建模不同时期的选址偏好变化;

- **跨城市迁移:** 研究选址模式的跨城市迁移学习, 实现冷启动城市的快速预测;

- **可解释性增强:** 通过注意力可视化和特征重要性分析, 提供选址决策的可解释性支持;

- **在线学习机制:** 设计增量学习算法, 使模型能够持续适应新的选址数据。

本研究为商业智能选址提供了新的技术路径, 对零售企业的扩张决策具有重要的实践价值。

## 参考文献

- [1] Church, R. L. (2009). Business site selection, location analysis, and GIS. John Wiley & Sons.
- [2] Roig-Tierno, N., Baviera-Puig, A., Buitrago-Vera, J., & Mas-Verdú, F. (2013). The retail site location decision process using GIS and the analytical hierarchy process. *Applied Geography*, 40, 191-198.
- [3] Karamshuk, D., Noulas, A., Scellato, S., Nicosia, V., & Mascolo, C. (2013). Geo-spotting: mining online location-based services for optimal retail store placement. In *Proceedings of KDD* (pp. 793-801).
- [4] Liu, Y., Sui, Z., Kang, C., & Gao, Y. (2016). Uncovering patterns of inter-urban trip and spatial interaction from social media check-in data. *PloS one*, 11(1), e0146291.
- [5] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [6] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph attention networks. In *ICLR*.
- [7] Huff, D. L. (1963). A probabilistic analysis of shopping center trade areas. *Land economics*, 39(1), 81-90.
- [8] Christaller, W. (1966). Central places in southern Germany. Prentice-Hall.
- [9] Weber, A. (1909). Theory of the Location of Industries. University of Chicago Press.
- [10] Yan, B., Janowicz, K., Mai, G., & Gao, S. (2017). From ITDL to Place2Vec: Reasoning about place type similarity and relatedness by learning embeddings from augmented spatial contexts. In *SIGSPATIAL* (pp. 1-10).
- [11] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- [12] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*.
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.
- [14] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *ICML*.
- [15] Zhang, Z., Wang, C., Gan, C., Sun, S., & Wang, M. (2019). Automatic modulation classification using CNN-LSTM based dual-stream structure. *IEEE Transactions on Vehicular Technology*, 69(11), 13521-13531.
- [16] Liu, F., Zhang, D., Shen, J., & Ma, J. (2020). KNN-based representation learning for graph classification. In *IJCAI*.
- [17] Wang, Y., Dong, L., Li, Y., & Zhang, H. (2021). Multitask feature learning approach for knowledge graph enhanced recommendations with RippleNet. *IEEE Transactions on Knowledge and Data Engineering*, 33(8), 3235-3248.
- [18] Goldberg, D. W. (2013). Advances in geocoding research and practice. *Transactions in GIS*, 17(1), 63-66.
- [19] Zhang, J., & Chow, C. Y. (2018). Point-of-interest recommendations in location-based social networks. *SIGSPATIAL Special*, 10(1), 19-25.
- [20] Li, Y., Zheng, Y., Zhang, H., & Chen, L. (2019). Traffic prediction in a bike-sharing system. In *SIGSPATIAL*.
- [21] Chen, W., Huang, Z., Wu, F., Zhu, M., Guan, H., & Maciejewski, R. (2020). VAUD: A visual analysis approach for exploring spatio-temporal urban data. *IEEE Transactions on Visualization and Computer Graphics*, 26(11), 3326-3338.

- [22] Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., & Wang, M. (2021). SocialGCN: An efficient graph convolutional network based model for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*.
- [23] Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., & Smola, A. J. (2017). Deep sets. In *NeurIPS*.
- [24] Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*.
- [25] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [26] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. In *ICLR*.