**算法选型**

* **初始聚类**：
  + **ST‑DBSCAN**（时空 DBSCAN）
    - 同时考虑空间（POI 坐标）与序列（访问顺序/时间戳）相似性
    - 无需预设簇数，能发现任意形状簇
* **深度表示**：
  + **行程序列 Transformer（“TripBERT”）**
    - 基于 HuggingFace BERT，将城市/POI ID 作为 Token
    - 微调任务：Masked POI、下一个 POI 预测

**2. 特征构造**

1. **城市–POI 嵌套序列**
   * 序列格式（示例）：

csharp

复制编辑

[Day1.City, Day1.POI1, Day1.POI2, …, Day2.City, Day2.POI3, …]

1. **地理坐标序列**
   * 每个 POI 的经纬度 (lat, lon) 按访问顺序排列
2. **访问时间戳/停留时长**
   * 对应每个 POI 记录访问开始时间或停留分钟数
3. **辅助特征**
   * 天数、预算区间、主题标签、一日内城市切换次数

**3. 表征提取与融合**

1. **预训练嵌入**
   * **POI Word2Vec**：用所有行程语料训练 Skip‑gram，得到 POI 向量
2. **Transformer 编码**
   * 输入：城市–POI 嵌套序列 + 位置编码 + 时间编码
   * 输出：CLS 向量作为整条行程深度表示
3. **混合相似度**
   * 序列距离：d\_seq = DTW(cities+POIs)
   * 嵌入距离：d\_emb = 1 – cosine(TripBERT\_CLS)
   * **最终距离**：d = α·normalize(d\_seq) + (1–α)·normalize(d\_emb)，α≈0.5

**4. 系统流程**

1. **数据预处理**
   * 标准化 POI/城市 ID，并填充缺失访问时间。
2. **特征生成**
   * 构建嵌套序列、坐标序列、时间序列和辅助特征。
3. **嵌入微调**
   * 在旅游行程语料上，用 Masked-POI + Next-POI 任务微调 BERT。
4. **聚类执行**
   * 对每条行程：
     1. 计算 d\_seq 与 d\_emb
     2. 归一化并合成距离矩阵
     3. 运行 ST‑DBSCAN，生成初步簇
5. **簇内去重筛选**
   * 每簇保留**簇中心**（最小平均距离行程）
   * 若簇中心属性单一，按“预算×标签多样性”二次排序，选取天数、主题各异的候选
6. **多样性保证**
   * 对最终保留集，检查国家/省/市覆盖度、价格分布、行程长度，必要时从下一个近邻补充

**7.1 方案1：混合相似度与聚类方法**

* **描述**：该方法结合多种专业相似度度量，形成对线路相似性的全面理解，然后通过聚类步骤对重复线路进行分组并选择代表。
  1. **特征工程**：实施第3节中详述的特征工程步骤，生成POI序列嵌入、多模态（文本+图像）嵌入和数值特征。
  2. **多指标相似度计算**：
     + 计算线路poi列表（POI序列）的DTW距离，以捕捉灵活的时空对齐。
     + 计算途径城市、途径国家、行程特色/标签集合的Jaccard相似度。
     + 计算组合多模态嵌入（来自标题、标签、图像、POI序列嵌入）的余弦相似度，以捕捉语义和视觉相似性。
     + 将这些单独的相似度分数组合成一个单一的复合相似度分数。对于演示，初始权重可以根据经验进行调整（例如，基于POI序列相对于标题的感知重要性）。
  3. **聚类去重**：在复合相似度矩阵上应用聚类算法（例如层次聚类或DBSCAN）。层次聚类允许可视化相似度树状图并选择截断阈值，而DBSCAN可以识别不同密度的聚类，无需预先指定k。每个聚类代表一组高度相似的线路。
  4. **代表选择**：从每个已识别的聚类中选择一个代表性线路。标准可以包括：最旧的线路（最早创建）、具有最高内部质量分数的线路，或具有最全面元数据的线路。
  5. **约束聚类（集成）**：对于特定的属性总和约束，可以在步骤3中识别的聚类上应用修改后的K-means（如第6节所述），或在可行的情况下将其集成到主要聚类的距离度量中。
* **优点**：
  1. 通过集成多样化的数据类型和专业指标，提供了全面而细致的相似度评估。
  2. 聚类本身处理重复组，并直接支持多样性要求。
  3. 灵活可调，适用于演示阶段的迭代优化。
* **缺点**：
  1. 在集成LSH之前，生成完整的成对相似度矩阵对于5万多条线路来说计算量可能很大（O(N^2)）。
  2. 在没有标注真实数据的情况下，确定组合不同相似度指标的最佳权重可能具有主观性。
  3. 精确集成“属性总和”约束可能需要自定义算法修改。

**7.2 方案2：以嵌入为中心结合LSH的方法**

* **描述**：该解决方案优先利用密集向量嵌入的强大功能进行线路的整体表示，并利用局部敏感哈希（LSH）实现高效的可伸缩性。
  1. **全面线路嵌入**：为每条线路生成一个单一的高维向量嵌入。这通过融合（例如，串联后接神经网络层，或注意力机制）所有工程特征来实现：POI序列嵌入 、多模态嵌入（文本+图像） 和数值特征（

天数、价格区间、聚合的建议游玩时间）。注意力机制 可以学习这些组合特征的最佳权重。

* 1. **LSH高效相似度搜索**：对这些高维线路嵌入应用局部敏感哈希（LSH）。LSH将通过将相似嵌入哈希到相同的桶中，快速识别出一小组

*候选*重复对。这大大减少了所需进行完整相似度计算的数量。

* 1. **精确相似度计算与聚类**：对于LSH过滤后的候选对，计算其嵌入之间的精确余弦相似度（或欧几里得距离）。然后，应用聚类算法（例如DBSCAN，它非常适合高维空间中的基于密度的聚类）来形成最终的重复组。
  2. **代表选择**：根据定义的标准从每个聚类中选择一个代表性线路。
  3. **约束聚类（集成）**：属性总和约束可以在聚类阶段纳入。可以对嵌入应用约束K-means变体（如第6节所述），确保聚类遵守建议游玩时间的总和约束 。这通常涉及修改聚类算法的距离度量或分配步骤。
* **优点**：
  1. 由于LSH的存在，对于大型数据集具有高度可伸缩性，适用于5万多条线路和每日增量。
  2. 嵌入捕捉丰富、抽象的表示，包括语义、视觉、序列和数值信息。
  3. 利用最先进的机器学习技术（多模态嵌入、LSH）。
* **缺点**：
  1. 训练/生成高质量嵌入需要大量的计算资源和专业知识。
  2. LSH是一种近似方法；可能会遗漏少量真正的重复项（假阴性），需要仔细调整参数。
  3. 调整LSH参数（例如，哈希函数数量k，哈希表数量L）对于平衡准确性和速度至关重要 。

**7.3 方案3：基于图的社区检测**

* **描述**：该方法将线路之间的关系建模为图，然后应用社区检测算法来查找相似线路的聚类。
  1. **相似度图构建**：
     + 计算所有线路之间的成对相似度分数（使用方案1中的复合相似度，或方案2中的嵌入相似度）。
     + 构建一个图，其中每条线路都是一个节点。相似度分数高于某个阈值的线路之间连接边，边权重代表相似度分数。这创建了一个稀疏图，重点关注强关系。
  2. **社区检测**：应用社区检测算法（例如Louvain、Infomap或Markov聚类算法 ）来识别密集子图（社区）。这些社区代表高度相似线路的聚类。社区检测旨在将节点划分为聚类，其中内部连接强于外部连接 。
  3. **代表选择**：从每个已识别的社区中选择一个代表。
  4. **约束聚类（集成）**：将“属性总和”约束直接集成到标准社区检测算法中具有挑战性。潜在的方法包括：
     + **图修改**：通过惩罚那些如果合并将违反属性总和约束的边来修改图的构建。
     + **专业图划分**：探索更高级的图划分算法，这些算法旨在处理容量约束（尽管这些算法通常比通用社区检测更复杂且不常见）。
     + **后处理**：应用后处理步骤，对违反属性总和约束的社区进行进一步分裂或细化，但这可能会损害原始的社区结构。
* **优点**：
  1. 为线路之间的复杂关系提供了自然直观的表示。
  2. 社区检测算法对不规则聚类形状具有鲁棒性，可以识别层次结构或重叠结构 。
  3. 利用了成熟的网络科学原理 。
* **缺点**：
  1. 图构建（特别是对于密集图）如果未优化（例如，通过使用稀疏图），则对于5万多个节点来说，内存和计算量可能很大。
  2. 直接纳入“属性总和”约束并非大多数标准社区检测算法固有的功能，可能需要大量的自定义开发或近似。
  3. 某些社区检测算法的计算复杂度可能很高，尽管存在许多优化的库。

 **阶段1（核心相似度与去重）**：

* **重点**：建立基础去重管道。
* **实施**：从最关键的特征子集（例如POI序列、城市、标题）开始。对序列实施DTW，对集合实施Jaccard，并对标题实施基本的文本嵌入（例如Sentence-Transformers）。将这些组合成一个简单的加权复合相似度分数。应用层次聚类或DBSCAN来演示重复项的分组。
* **成果**：一个可工作的原型，展示如何识别和分组相似线路，以及如何选择代表。

 **阶段2（高级嵌入与可伸缩性）**：

* **重点**：提高相似度准确性并解决全数据集的可伸缩性问题。
* **实施**：集成多模态嵌入（用于图像和文本）和上下文感知POI嵌入（CatEM）。在组合的高维线路嵌入上实施LSH，以加速对整个5万多条数据集和新传入数据的相似度搜索。
* **成果**：一个更准确、更可伸伸缩的去重系统，能够处理数据的全部复杂性和体量。

 **阶段3（约束聚类集成）**：

* **重点**：解决空间聚类与属性总和控制的特定要求。
* **实施**：集成修改后的K-means算法（或如果与演示环境兼容，探索空间约束多变量聚类工具），以便在主要聚类期间或之后应用建议游玩时间总和约束。
* **成果**：演示系统如何对结果聚类强制执行特定的业务规则。

 **阶段4（评估与优化）**：

* **重点**：量化性能并优化参数。
* **实施**：创建一个小规模的人工标注真实数据集进行评估。使用第8节中定义的指标评估去重有效性和多样性保留。迭代调整相似度阈值、聚类参数，并可能调整特征权重。
* **成果**：系统性能的数据驱动验证以及进一步改进领域的识别。