Technical Report on Beijing Resident Daily Travel Trajectory Generation System

I. System Overview

This system is a simulator of Beijing residents' daily travel trajectories based on a Large Language Model (LLM). It integrates resident attributes (age, occupation, family status, budget, etc.) and Beijing POI (Point of Interest) data to generate logical and personalized daily travel trajectories, and supports quality assessment of the generated trajectories. The program will sequentially: read resident attributes, find nearby POIs for each resident, call the Tongyi Qianwen model to generate personalized trajectories, and aggregate all trajectories.

1. System Operation Process

1.1 Data Preparation:

Collect Beijing POI data and generate simulated resident attribute data, etc.

1.2 Trajectory Generation:

Generate personalized travel trajectories based on resident attributes and nearby POIs using an LLM.

1.3 Trajectory Evaluation:

Evaluate the quality of generated trajectories from three dimensions: visit distribution (KL divergence), movement radius, and trip chain length.

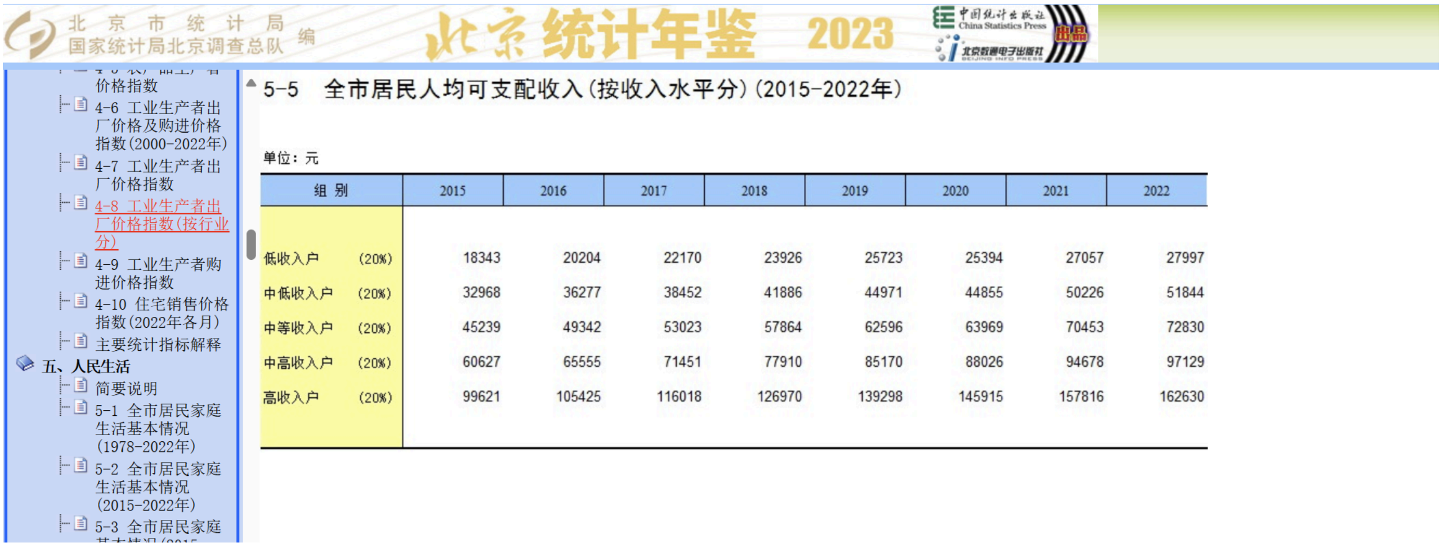
2. Program File Description:

- `agent.py` - Main program, responsible for loading data and generating resident trajectories.
- `evaluate.py` - Evaluation script, calculates various metrics for the generated trajectories.
- `get_beijing_poi.py` - POI data collection script (fetches data from the Amap API).
- `get_user.py` - User Data generation script
- `prompt_template.txt` - LLM prompt template.
- `residents_100.csv` - Attribute data for 100 simulated residents.
- `trajectories.json` - Pre-generated trajectory results file.

II. Data Sources

1. User Data Construction:

Due to privacy protection reasons, we did not collect user data strongly correlated with travel trajectories. Therefore, we generated one hundred virtual residents based on the attribute distribution of Beijing residents. The attributes of the virtual residents follow real statistical data distributions, while ensuring data consistency for each individual resident.
Sources for Beijing resident data:
https://usa.ipums.org: China 2020 Population Census 1% Microdata (IPUMS International 2020 China 1% sample)
https://tjj.beijing.gov.cn: 《Beijing Statistical Yearbook 2023》



价格指数
4-6 工业生产者出厂价格及购进价格指数(2000-2022年)
4-7 工业生产者出厂价格指数
4-8 工业生产者出厂价格指数(按行业分)
4-9 工业生产者购进价格指数
4-10 住宅销售价格指数(2022年各月)
主要统计指标解释
五、人民生活
简要说明
5-1 全市居民家庭生活基本情况(1978-2022年)
5-2 全市居民家庭生活基本情况(2015-2022年)
5-3 全市居民家庭

5-5 全市居民人均可支配收入(按收入水平分)(2015-2022年)

单位：元

组 别	2015	2016	2017	2018	2019	2020	2021	2022
低收入户 (20%)	18343	20204	22170	23926	25723	25394	27057	27997
中低收入户 (20%)	32968	36277	38452	41886	44971	44855	50226	51844
中等收入户 (20%)	45239	49342	53023	57864	62596	63969	70453	72830
中高收入户 (20%)	60627	65555	71451	77910	85170	88026	94678	97129
高收入户 (20%)	99621	105425	116018	126970	139298	145915	157816	162630

2. POI List Acquisition:

Applied for the Amap API, obtained the main POI list of Beijing through place search.

3. Real POI Visit Distribution of Beijing Residents:

Data obtained from existing research.
https://ir.bjtu.edu.cn/handle/10364/9359:
《Research on Spatio-temporal Characteristics of Beijing Residents' Travel Based on Subway Card Swipes》

4. Real Average Movement Radius of Beijing Residents:

Used the average commuting distance of Beijing residents on workdays as the average movement radius on workdays.
North News Network - News Center: news.enorth.com.cn

5. Workday Trip Chain Length of Beijing Residents:

《Impact Mechanism of Transportation Carbon Emissions in Beijing Based on Resident Travel Behavior》 Ma Jing, Chai Yanwei, Liu Zhilin - Acta Geographica Sinica, 2011

III. Overall Design Concept

1. Virtual User Generation Logic

1.1 Data Preparation and Initialization:

Use pandas to read two external data files:
beijing_pois_combined.csv: Contains Beijing POI (Point of Interest) information, used to provide spatially realistic home addresses;
full_weights.csv: Contains distribution weights for attributes such as age, gender, occupation, family role, budget level.
Ensure the reproducibility of generation results by setting a random seed (random.seed(42)).

1.2 Address Validity Filtering:

Exclude POI types unsuitable as home addresses (e.g., shopping malls, parks, hospitals, etc.) from the POI data to build a valid address pool `valid_locations` .

1.3 Stratified Weighted Sampling:

Perform weighted random sampling for each record according to the weight distribution in `full_weights.csv` , ensuring the generated user attributes conform to the preset socio-statistical distribution.
For each selected record, parse the age bracket (e.g., "20-30"), randomly generate a specific age within that range, and randomly assign a home address from the valid address pool.

1.4 Record Construction and Output:

Assign a unique ID to each virtual user and build a record containing the following attributes:
id, age, gender, occupation, family_role, budget_level, home
Combine all records into a DataFrame and export it as `residents_100.csv` .

2. Agent Logic

2.1 System Initialization and Data Loading:

Use `langchain_community.chat_models.ChatTongyi` to access the Tongyi Qianwen large model (`qwen-max-latest`), set the temperature to 0.3 to control generation diversity.
Read two key data sources:
beijing_pois_combined.csv: Beijing Point of Interest (POI) dataset, containing names, types, coordinates, etc.;
residents_100.csv: Virtual resident attribute table, containing age, occupation, budget, family role, home coordinates, etc.
Load the natural language prompt template from the external file `prompt_template.txt` to construct LLM input.

2.2 Trajectory Generation Logic:

For each resident,
Get the list of nearby POIs;
Concatenate POI names into a string, truncate to the first 5000 characters to avoid context overflow;
Use the prompt template to construct the input, passing in resident attributes and nearby POI information;
Call the large model to generate a daily itinerary trajectory that conforms to their social attributes (age, occupation, budget, etc.).

2.3 Asynchronous Execution and Caching Mechanism:

Use `asyncio` to asynchronously handle multi-resident trajectory generation tasks, improving efficiency;
Implement a local JSON caching mechanism: if a resident's trajectory already exists, read it directly to avoid repeated generation;
Finally, save all trajectories uniformly to the `trajectories.json` file.

3. Evaluation Logic

3.1 Evaluation Metrics Overview:

Visit Distribution (KL Divergence): Compares the difference between the generated trajectory's POI visit distribution and the real one.
Average Movement Radius: The maximum activity range of a resident within a day (kilometers).
Average Trip Chain Length: The number of POIs in a trajectory.

3.2 Visit Distribution (KL Divergence):

Metric Significance:
Evaluates whether the distribution of POI types visited in the generated trajectories is consistent with the real-world resident visit preferences.
Prior Distribution:
A known POI visit probability distribution derived from actual observations or research is used as a benchmark. In this evaluation, this distribution is defined as: {'Office Building':0.437, 'Shopping Mall':0.211, 'Residential Area':0.297, 'Other':0.055}.
Simulated Distribution:
Step 1: Parse all generated trajectories, extract every visited POI name.
Step 2: Map the POI names to their corresponding types (e.g., "Shopping Mall") using a pre-built mapping dictionary.
Step 3: Count the frequency of occurrence of all POI types and normalize it to obtain the simulated distribution.
Calculation:
Use Kullback-Leibler (KL) Divergence to calculate the difference between the simulated distribution Q and the prior distribution P. A KL divergence value closer to 0 indicates that the two distributions are more similar, meaning the visit preferences of the generated trajectories are more realistic.

$$D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

3.3 Average Movement Radius:

Metric Significance:
Quantifies the maximum spatial range involved in a resident's single-day activities, testing whether the generated trajectories are reasonably constrained within the typical distances of daily commuting and activities.
Coordinate Conversion:
For each POI in a trajectory, obtain its latitude and longitude coordinates through the mapping dictionary and convert them into a tuple of floats.

Distance Calculation:
For all pairwise combinations of POIs in a single trajectory, calculate the spherical distance between them using the Haversine formula.
Trajectory Radius:
Define the movement radius of a single trajectory as the maximum distance between any two points in that trajectory.
Aggregate Analysis:
Calculate the average of all resident trajectory radii as the average movement radius of the entire simulated population.

3.4 Average Trip Chain Length:

Metric Significance:
Measures the activity level of residents' daily travel, i.e., the average number of unique locations visited per day.
Parse Trajectory:
Parse each generated trajectory, converting it into a list of POI locations.
Count:
Directly calculate the length of each trajectory list, which is the daily trip chain length for that resident.
Aggregate Analysis:
Calculate the average of all residents' trip chain lengths to obtain the typical number of daily activities at the group level.

IV. Operation Effectiveness and Result Verification

1. Operation Logs

1.1 Agent Simulation Operation Log

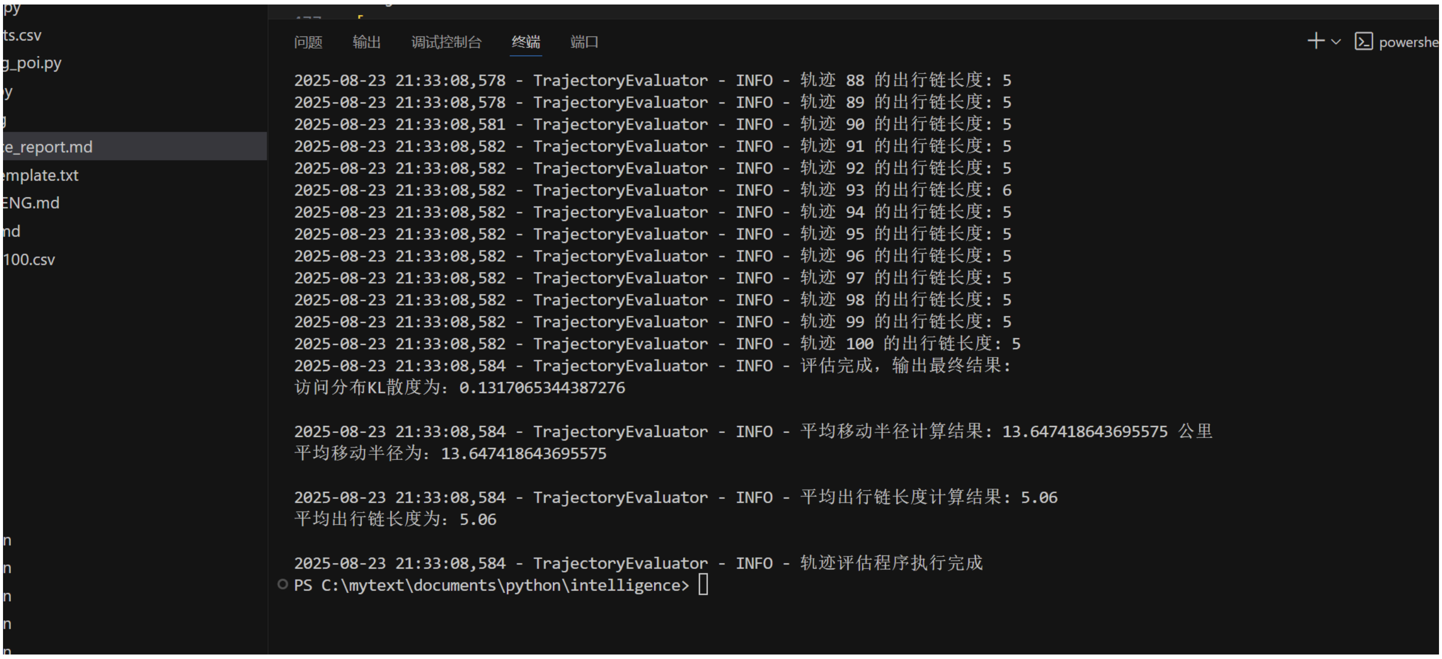
```
2025-08-23 20:55:04,505 - TrajectoryGenerator - INFO - Trajectory generation program started
2025-08-23 20:55:04,510 - TrajectoryGenerator - INFO - Started processing trajectory generation tasks for 100 residents
2025-08-23 20:55:04,511 - TrajectoryGenerator - INFO - Trajectory for resident 1 does not exist, starting generation
2025-08-23 20:55:04,511 - TrajectoryGenerator - INFO - Starting trajectory generation for resident 1 (Age: 30, Occupation: Blue-collar)
2025-08-23 20:55:04,511 - TrajectoryGenerator - INFO - Searching for POIs within 10 km of coordinates 116.437273,39.894630
2025-08-23 20:55:04,598 - TrajectoryGenerator - INFO - Found 1331 nearby POIs
2025-08-23 20:55:04,651 - TrajectoryGenerator - INFO - Calling large language model to generate trajectory...
2025-08-23 20:55:08,347 - TrajectoryGenerator - INFO - Trajectory for resident 1 generated successfully
2025-08-23 20:55:08,349 - TrajectoryGenerator - INFO - Trajectory for resident 1 saved to traj_1.json
2025-08-23 20:55:08,351 - TrajectoryGenerator - INFO - Trajectory for resident 2 does not exist, starting generation
2025-08-23 20:55:08,352 - TrajectoryGenerator - INFO - Starting trajectory generation for resident 2 (Age: 40, Occupation: Senior White-coll
2025-08-23 20:55:08,352 - TrajectoryGenerator - INFO - Searching for POIs within 10 km of coordinates 116.438142,39.873809
...
2025-08-23 21:01:45,338 - TrajectoryGenerator - INFO - Trajectory for resident 98 does not exist, starting generation
2025-08-23 21:01:45,346 - TrajectoryGenerator - INFO - Starting trajectory generation for resident 98 (Age: 28, Occupation: Mid-level White-
2025-08-23 21:01:45,347 - TrajectoryGenerator - INFO - Searching for POIs within 10 km of coordinates 116.664867,40.133987
2025-08-23 21:01:45,486 - TrajectoryGenerator - INFO - Found 54 nearby POIs
2025-08-23 21:01:45,491 - TrajectoryGenerator - INFO - Calling large language model to generate trajectory...
2025-08-23 21:01:48,897 - TrajectoryGenerator - INFO - Trajectory for resident 98 generated successfully
2025-08-23 21:01:48,905 - TrajectoryGenerator - INFO - Trajectory for resident 98 saved to traj_98.json
2025-08-23 21:01:48,918 - TrajectoryGenerator - INFO - Trajectory for resident 99 does not exist, starting generation
2025-08-23 21:01:48,925 - TrajectoryGenerator - INFO - Starting trajectory generation for resident 99 (Age: 91, Occupation: Retiree)
2025-08-23 21:01:48,925 - TrajectoryGenerator - INFO - Searching for POIs within 10 km of coordinates 116.312263,39.882620
2025-08-23 21:01:49,085 - TrajectoryGenerator - INFO - Found 935 nearby POIs
2025-08-23 21:01:49,145 - TrajectoryGenerator - INFO - Calling large language model to generate trajectory...
2025-08-23 21:01:52,838 - TrajectoryGenerator - INFO - Trajectory for resident 99 generated successfully
2025-08-23 21:01:52,843 - TrajectoryGenerator - INFO - Trajectory for resident 99 saved to traj_99.json
2025-08-23 21:01:52,846 - TrajectoryGenerator - INFO - Trajectory for resident 100 does not exist, starting generation
2025-08-23 21:01:52,850 - TrajectoryGenerator - INFO - Starting trajectory generation for resident 100 (Age: 27, Occupation: Service Industr
2025-08-23 21:01:52,852 - TrajectoryGenerator - INFO - Searching for POIs within 10 km of coordinates 116.705862,40.325562
2025-08-23 21:01:53,010 - TrajectoryGenerator - INFO - Found 19 nearby POIs
2025-08-23 21:01:53,015 - TrajectoryGenerator - INFO - Calling large language model to generate trajectory...
2025-08-23 21:01:57,807 - TrajectoryGenerator - INFO - Trajectory for resident 100 generated successfully
2025-08-23 21:01:57,811 - TrajectoryGenerator - INFO - Trajectory for resident 100 saved to traj_100.json
2025-08-23 21:01:57,812 - TrajectoryGenerator - INFO - All resident trajectories processed, total 100
2025-08-23 21:01:57,828 - TrajectoryGenerator - INFO - All trajectories saved to trajectories.json, program execution completed
```

1.2 Evaluation File Run Log

```
2025-08-23 21:33:08,463 - TrajectoryEvaluator - INFO - Started loading trajectory data...
2025-08-23 21:33:08,463 - TrajectoryEvaluator - INFO - Successfully loaded 100 trajectory data entries
2025-08-23 21:33:08,464 - TrajectoryEvaluator - INFO - Started loading POI data...
2025-08-23 21:33:08,472 - TrajectoryEvaluator - INFO - Successfully loaded 2656 POI data entries
2025-08-23 21:33:08,475 - TrajectoryEvaluator - INFO - Started loading resident data...
2025-08-23 21:33:08,475 - TrajectoryEvaluator - INFO - Successfully loaded 100 resident data entries
2025-08-23 21:33:08,475 - TrajectoryEvaluator - INFO - Creating POI name to type mapping dictionary...
2025-08-23 21:33:08,480 - TrajectoryEvaluator - INFO - Started calculating visit distribution KL divergence...
2025-08-23 21:33:08,480 - TrajectoryEvaluator - INFO - Real distribution: {'Office Building': 0.437, 'Shopping Mall': 0.211, 'Residential Ar
2025-08-23 21:33:08,484 - TrajectoryEvaluator - INFO - Extracted 506 POI visit records from trajectories
2025-08-23 21:33:08,486 - TrajectoryEvaluator - INFO - Simulated distribution: {'Office Building': 0.1324110671936759, 'Shopping Mall': 0.20
2025-08-23 21:33:08,490 - TrajectoryEvaluator - INFO - KL divergence calculation result: 0.1317065344387276
2025-08-23 21:33:08,490 - TrajectoryEvaluator - INFO - Started calculating movement radius...
2025-08-23 21:33:08,490 - TrajectoryEvaluator - INFO - Creating POI name to location mapping dictionary...
2025-08-23 21:33:08,490 - TrajectoryEvaluator - INFO - Calculating movement radius for all trajectories...
2025-08-23 21:33:08,491 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 1/100
2025-08-23 21:33:08,491 - TrajectoryEvaluator - INFO - Movement radius for trajectory 1: 12.92971675203858 km
2025-08-23 21:33:08,492 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 2/100
2025-08-23 21:33:08,492 - TrajectoryEvaluator - INFO - Movement radius for trajectory 2: 15.74051424694059 km
2025-08-23 21:33:08,492 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 3/100
2025-08-23 21:33:08,493 - TrajectoryEvaluator - INFO - Movement radius for trajectory 3: 19.978223345125166 km
...
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 97/100
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Movement radius for trajectory 97: 15.758279282237998 km
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 98/100
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Movement radius for trajectory 98: 8.780016372612456 km
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 99/100
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Movement radius for trajectory 99: 16.655909962529556 km
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Calculating movement radius for trajectory 100/100
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Movement radius for trajectory 100: 2.0935039333412027 km
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Started calculating trip chain length...
2025-08-23 21:33:08,564 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 1: 5
2025-08-23 21:33:08,567 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 2: 5
...
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 97: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 98: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 99: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - Trip chain length for trajectory 100: 5
2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - Evaluation completed, outputting final results:
Visit distribution KL divergence is: 0.1317065344387276
2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - Average movement radius calculation result: 13.647418643695575 km
Average movement radius is: 13.647418643695575
2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - Average trip chain length calculation result: 5.06
Average trip chain length is: 5.06
```

2. Results Summary

Evaluation Results:
Visit distribution KL divergence is: 0.1317065344387276
Average movement radius is: 13.647418643695575 km
Average trip chain length is: 5.06



```
2025-08-23 21:33:08,578 - TrajectoryEvaluator - INFO - 轨迹 88 的出行链长度: 5
2025-08-23 21:33:08,578 - TrajectoryEvaluator - INFO - 轨迹 89 的出行链长度: 5
2025-08-23 21:33:08,581 - TrajectoryEvaluator - INFO - 轨迹 90 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 91 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 92 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 93 的出行链长度: 6
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 94 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 95 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 96 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 97 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 98 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 99 的出行链长度: 5
2025-08-23 21:33:08,582 - TrajectoryEvaluator - INFO - 轨迹 100 的出行链长度: 5
2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - 评估完成，输出最终结果:
访问分布KL散度为: 0.1317065344387276

2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - 平均移动半径计算结果: 13.647418643695575 公里
平均移动半径为: 13.647418643695575

2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - 平均出行链长度计算结果: 5.06
平均出行链长度为: 5.06

2025-08-23 21:33:08,584 - TrajectoryEvaluator - INFO - 轨迹评估程序执行完成
PS C:\mytext\documents\python\intelligence> 
```