International Journal of Geo-Information

MDPI

Article

Trajectory Compression with Spatio-Temporal Semantic Constraints

Yan Zhou [1,2], Yunhan Zhang [2,*], Fangfang Zhang [1,3], Yeting Zhang [4] and Xiaodi Wang [2]

[1] The Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources, Shenzhen 518063, China; zhouyan_gis@uestc.edu.cn (Y.Z.)
[2] School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China; 202221070111@std.uestc.edu.cn
[3] Shenzhen Data Management Center of Planning and Natural Resources, Shenzhen 518040, China
[4] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China; zhangyeting@whu.edu.cn
* Correspondence: 202252070314@std.uestc.edu.cn

Abstract: Most trajectory compression methods primarily focus on geometric similarity between compressed and original trajectories, lacking explainability of compression results due to ignoring semantic information. This paper proposes a spatio-temporal semantic constrained trajectory compression method. It constructs a new trajectory distance measurement model integrating both semantic and spatio-temporal features. This model quantifies semantic features using information entropy and measures spatio-temporal features with synchronous Euclidean distance. The compression principle is to retain feature points with maximum spatio-temporal semantic distance from the original trajectory until the compression rate is satisfied. Experimental results show these methods closely resemble each other in maintaining geometric similarity of trajectories, but our method significantly outperforms DP, TD-TR, and CascadeSync methods in preserving semantic similarity of trajectories. This indicates that our method considers both geometric and semantic features during compression, resulting in the compressed trajectory becoming more interpretable.

Keywords: spatio-temporal trajectory; trajectory compression; geometric similarity; semantic similarity; information entropy

check for updates

Citation: Zhou, Y.; Zhang, Y.; Zhang, F.; Zhang, Y.; Wang, X. Trajectory Compression with Spatio-Temporal Semantic Constraints. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 212. https://doi.org/10.3390/ijgi13060212

Academic Editors: Wolfgang Kainz and Hartwig H. Hochmair

Received: 23 April 2024
Revised: 13 June 2024
Accepted: 15 June 2024
Published: 18 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Trajectory data, obtained by sampling the motion process of moving objects, includes the position, time, speed, and other relevant information of each sampling point [1]. The curve formed by these points according to time series is called a trajectory. The rapid development of Global Position System (GPS) and mobile terminal equipment has made it easier to obtain trajectory data but has also led to challenges such as large data volume and high redundancy. Trajectory data compression technology can effectively reduce the size of trajectory data while preserving the similarity between the original and compressed trajectories, thereby helping to alleviate the challenges associated with managing, analyzing, and visualizing large-scale trajectory data [2].

Most trajectory compression methods mainly focus on spatial features of trajectories, aiming to preserve geometric similarity between compressed and original trajectories [3–5]. Some studies have considered temporal dimension features of trajectories and optimized the selection of compressed trajectory feature points, achieving impressive spatio-temporal compression effects [1,2,6]. In addition to temporal and spatial features, trajectory data also contain valuable semantic information. For instance, analyzing trajectory speed can reveal whether a user is moving by transportation or walking, while mining Points of Interest (POI) along a trajectory can unveil a user's behavior patterns. However, most existing trajectory compression methods predominantly focus on spatio-temporal features

of trajectories and tend to overlook their semantic features. As a result, this often leads to a lack of explainability in the compression results of trajectories.

To address this, we propose trajectory compression with spatio-temporal semantic constraints. This method constructs a spatio-temporal semantic distance model that considers both spatio-temporal and semantic features of trajectories. When selecting trajectory points, our method considers not only the spatio-temporal morphological similarity of the compressed trajectory but also introduces information entropy to constrain the semantic similarity of trajectories. This ensures that the compressed trajectory retains maximum semantic similarity with the original trajectory. By incorporating semantic constraints, this method overcomes the problem of trajectory data lacking explainability due to semantic loss in traditional compression methods. As a result, it effectively supports deep-level trajectory analysis and mining.

The rest of this paper is organized as follows. Related works are discussed in Section 2. Section 3 presents the spatio-temporal semantic distance measuring model and the principle of the trajectory compression method. Experiments and analysis are described in Section 4. This paper ends with conclusions in Section 5.

2. Related Work

Trajectory data exhibit spatial, temporal, and semantic features. Based on the three main features that trajectory compression methods focus on, we summarize existing methods into three types: spatial-based trajectory compression, spatio-temporal-based trajectory compression, and semantic-based trajectory compression.

Spatial-based trajectory compression primarily considers spatial features when selecting trajectory feature points, resulting in the compressed trajectory exhibiting similarity to the original trajectory in terms of spatial geometric shape. Typical methods in this category include the Bellman method [7], the Douglas–Peucker (DP) method [8], and the DPhull method [9]. To address the distance threshold dependency in the DP method, Liu et al. proposed an automatic threshold Douglas–Peucker (ADP) method [10]. Tang et al. introduced the threshold change ratio to improve feature point selection in the DP method [11]. Huang et al. improved compression performance using GPU acceleration for the DP method and kernel density estimation (KDE) [12]. Additionally, Li et al. presented a trajectory compression method based on recurrent neural networks (RNN), utilizing a deep learning model to improve the compression rate by preserving the spatial dimension feature [13]. However, these spatial-based trajectory compression methods focus on the spatial positions of the trajectory, ignoring the temporal and semantic features, resulting only in the geometric similarity between compressed and original trajectories.

Spatio-temporal-based trajectory compression introduces the time-dimensional feature in addition to considering the spatial feature, resulting in the compressed trajectory having spatio-temporal similarity with the original trajectory. Typical methods include the Top–Down Time-Ratio (TD-TR) method [14], Opening Window Time Ratio (OPW-TR) [15], STTrace method [16], and Spatial Quality Simplification Heuristics (SQUISH) method [17]. Furthermore, Liu et al. proposed a trajectory compression method based on stopover points by combining speed and residence time [18]. Zhao et al. presented a frequent pattern-based road network trajectory compression method [19], which first conducts spatial compression based on spatial frequent patterns and then performs temporal compression on the identified spatial patterns. Zheng et al. discussed a trajectory compression framework based on spatio-temporal reference [20]. These trajectory compression methods improve the spatio-temporal similarity between the compressed and original trajectories. However, as semantics are not considered during compression, some trajectory points with important semantics may be discarded, resulting in compressed trajectories lacking interpretability.

Semantic features play a crucial role in trajectory data mining, contributing to trajectory explainability and facilitating behavioral pattern discovery in compressed trajectories. Richter et al. proposed semantic trajectory compression (STC), which identifies and describes relevant events along a trajectory, combining them into event blocks to form the

compressed trajectory [21]. Mirvahabi et al. introduced constraints on compression, comprehensively considering data diversity, underlying attributes, and semantic information [22]. Zhang et al. compressed trajectories by identifying vehicle movement modes, reflecting semantic information [23]. These methods target semantic features of trajectories for compression, yielding interpretable compressed trajectories. However, they primarily focus on semantic similarity of compressed trajectories rather than spatio-temporal similarity. Additionally, Gao et al. proposed a semantic compression method [24], achieving semantic compression at different granularities based on multi-resolution synchronization-based clustering. It supports multi-level semantic characterization of trajectory data.

In summary, existing trajectory compression methods often struggle to consider spatio-temporal and semantic features simultaneously, limiting deep trajectory mining and analysis. We propose a trajectory compression method with spatio-temporal semantic constraints, utilizing a spatio-temporal semantic distance model to select trajectory feature points. The compressed trajectory exhibits both spatio-temporal and semantic similarity with the original, which can provide interpretable compressed trajectories for trajectory mining and large-scale data visualization.

3. The Proposed Method

3.1. Principle

This paper proposes a trajectory compression method with spatio-temporal semantic constraints. It constructs a spatio-temporal semantic distance model using synchronous Euclidean distance and information entropy to integrate spatio-temporal and semantic features of trajectories. By selecting feature points with the largest spatio-temporal semantic distance, the compressed trajectory achieves both geometric and semantic similarity to the original trajectory. This ensures geometric similarity to the original while preserving semantic features, yielding interpretable compressed results. In many existing methods, users must set thresholds for compressed parameters, such as maximum error distance, maximum angle, maximum velocity, etc. However, users may lack the experience to determine appropriate thresholds. Instead, our method only requires setting a compression ratio for trajectory compression. To facilitate description, relevant definitions are introduced.

Definition 1. *Spatio-temporal trajectory. A spatio-temporal trajectory represents the continuous movement positions of a moving object over time. A spatio-temporal trajectory T can be expressed as Formula (1).*

$$T = \{p_1, p_2, p_3, \cdots, p_n\} \tag{1}$$

where trajectory point p_i $(i = 1, 2, \ldots n)$ *is defined as* $p_i = \{x_i, y_i, t_i, s_i\}$;$x_i$,$y_i$,$t_i$ *denote the longitude, latitude, and time of the trajectory point, respectively.*s_i *stands for semantic information carried by the trajectory point, such as speed, direction, Point of Interest (POI), etc.*

Definition 2. *Trajectory feature points. Trajectory feature points are key trajectory points reflecting specific spatio-temporal trajectory characteristics.*

Definition 3. *Approximate trajectory. The approximate trajectory is the trajectory composed of all trajectory feature points in temporal order.*

Definition 4. *Approximate trajectory segment. An approximate trajectory segment is the trajectory segment formed by connecting two adjacent trajectory points in the approximate trajectory.*

The proposed method works as follows: First, set the trajectory compression ratio based on user requirements. Second, initialize an approximate trajectory using the start and end points of the original trajectory. Next, calculate the upper limit on the number of points in the compressed trajectory based on the compression ratio. If the number of points in the approximate trajectory is less than this upper limit, measure the spatio-temporal semantic distance from each trajectory point to the approximate trajectory using a

spatio-temporal semantic distance model. Select the trajectory point with maximum spatio-temporal semantic distance as a trajectory feature point, add it to the approximate trajectory, and split the approximate trajectory into two segments at this point. Repeat this process for the two segments until the number of trajectory feature points in the approximate trajectory meets or exceeds the upper limit. Finally, output the compressed trajectory consisting of all trajectory feature points. The detailed steps are as follows:

Step 1: Input trajectory T and set compression ratio R;

Step 2: Initialize approximate trajectory $simtr = \{p_1, p_n\}$ with start and end points of T (denoted p_1 and p_n, respectively);

Step 3: Calculate the upper limit N' on compressed trajectory points using Formula (2).

$$N' = [(1 - R) \times N] \tag{2}$$

where N is the number of trajectory points in T;

Step 4: If the number of points in $simtr$ is less than N', proceed to the next step. Otherwise, go to step 10;

Step 5: Measure the spatio-temporal semantic distance STS_Dist from each trajectory point in T to $simtr$ using the spatio-temporal semantic distance model (see Section 3.2 for details);

Step 6: Select a trajectory point with maximum STS_Dist as a trajectory feature point;

Step 7: Add trajectory feature point to $simtr$ in temporal order of trajectory points;

Step 8: Split $simtr$ the newly added trajectory feature point into two approximate trajectory segments;

Step 9: Repeat steps 4–8 for each approximate trajectory segment until the number of trajectory feature points in $simtr$ meets or exceeds N';

Step 10: Output approximate trajectory composed of all trajectory feature points as compression result.

3.2. Spatio-Temporal Semantic Distance Model

The core of the trajectory compression method is identifying trajectory feature points that maximally reflect the spatio-temporal and semantic characteristics of the original trajectory, ensuring spatio-temporal and semantic similarity between the original and compressed trajectories. Traditional trajectory compression methods mainly focus on geometric similarity and select feature points based on distance models such as Euclidean and Hausdorff distance. Some studies incorporate temporal factors using synchronous Euclidean distance. However, these distance models do not consider semantic features of trajectories, risking discarding points with significant semantic information, resulting in compressed results lacking explainability. This paper proposes a spatio-temporal semantic distance model, introducing information entropy to measure semantic features of trajectories and using synchronous Euclidean distance to measure spatio-temporal features. This method effectively constrains feature point selection from both semantic and spatio-temporal aspects, preventing the loss of semantically significant points during compression.

The spatio-temporal semantic distance from a trajectory point p_i to the trajectory T is denoted as $STS_Dist(p_i, T)$, which is calculated using Formula (3).

$$STS_Dist(p_i, T) = w_k \times \frac{ST_Dist(p_i, T)}{ST_Dist_{\max}} + (1 - w_k) \times \frac{Semantic_Dist(p_i, T)}{Semantic_Dist_{\max}} \tag{3}$$

where $ST_Dist(p_i, T)$ is the spatio-temporal distance between the trajectory point p_i and trajectory T, $ST_Dist_{\max}$ is the maximum value of the spatio-temporal distance among all trajectory points, and the ratio between them represents the normalized spatio-temporal distance. Similarly, $Semantic_Dist(p_i, T)$ denotes the semantic distance between p_i and T; $Semantic_Dist_{\max}$ is the maximum semantic distance among all trajectory points, and the ratio between the two values indicates the normalized semantic distance. The weights w_k and $1 - w_k$ are the spatio-temporal distance weight and the semantic distance weight,

respectively. $STS_Dist(p_i, T)$ couples the spatio-temporal distance and semantic distance to ensure the maximum preservation of spatio-temporal semantic similarity between the compressed trajectory and the original trajectory.

3.2.1. Spatio-Temporal Distance Measurement

The spatio-temporal distance $ST_Dist(p_i, T)$ is measured using the synchronous Euclidean distance (SED) [14].

As shown in Figure 1, $T = \{p_m, p_{m+1}, \cdots, p_i, \cdots, p_{k-1}, p_k\}$ denotes a trajectory; $p_i{}'(x_i{}', y_i{}', t_i)$ is the spatio-temporal projection of a trajectory point $p_i(x_i, y_i, t_i)$ onto the approximate trajectory $simtr = \{p_m, p_k\}$. The position of $p_i{}'$ can be calculated from $p_m(x_m, y_m, t_m)$ and $p_k(x_k, y_k, t_k)$, as detailed, using Formulas (4) and (5).

$$x_i{}' = x_m + \frac{t_i - t_m}{t_k - t_m}(x_k - x_m) \tag{4}$$

$$y_i{}' = y_m + \frac{t_i - t_m}{t_k - t_m}(y_k - y_m) \tag{5}$$

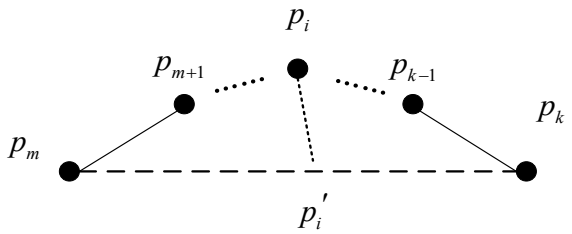

Figure 1. Synchronous Euclidean distance.

Then, the SED can be calculated using the position of p_i and $p_i{}'$, as shown in Formula (6).

$$SED = \sqrt{(x_i - x_i{}')^2 + (y_i - y_i{}')^2} \tag{6}$$

SED considers both the spatial and temporal information of trajectory points, in contrast to the Euclidean distance, which only considers spatial position. Thus, SED is more appropriate for measuring the spatio-temporal distance of a trajectory.

3.2.2. Semantic Distance Measurement

In this paper, we propose introducing information entropy as a metric to measure the semantic distance $Semantic_Dist(p_i, T)$ between trajectory points and a trajectory. Information entropy [25], a fundamental concept in information theory, quantifies the degree of order within a system. A more orderly system has a lower entropy value and vice versa. If a trajectory point exhibits higher semantic similarity with points on the approximate trajectory, adding that trajectory point to the approximate trajectory results in a more orderly new approximate trajectory. This implies that the information entropy of the approximate trajectory with the added trajectory point is smaller, and vice versa is larger. The calculation of information entropy is shown as Formula (7).

$$H(X) = -\sum P(x_i) \log_2 P(x_i), i = 1, 2, \cdots, n \tag{7}$$

where X denotes a random variable that takes the value of $(x_1, x_2, \cdots, x_n)$; $P(x_i)$ is the occurrence probability of x_i, and $\sum P(x_i) = 1$.

Given a trajectory T with m-dimensional semantics $S = (A_1, A_2, \cdots, A_m)$, where $A_i (i = 1, 2, \ldots m)$ denotes the ith dimensional semantic item, we define the semantic distance between a trajectory point p_i and T as the change in the trajectory information entropy caused by adding p_i to T. The new trajectory T' is obtained by adding p_i to T has

k dimensional semantics $S' = (A_1', A_2', \cdots, A_k')$; $m \le k \le m+1$. Then, the semantic distance between p_i and T is expressed using Formula (8).

$$Semantic_Dist(p_i, T) = \left| H(T') - H(T) \right| \tag{8}$$

where $|\cdot|$ denotes absolute value; $H(T)$ and $H(T')$ is the information entropy of trajectory T and T', respectively, which can be obtained using Formula (9):

$$\begin{cases} H(T) = \sum_{i=1}^{m} w_i \times H(A_i) \\ H(T') = \sum_{i=1}^{k} w_i \times H(A_i') \end{cases} \tag{9}$$

where $H(A_i)$ represents the information entropy of the semantic item A_i in trajectory T; w_i denotes the weight of the ith dimensional semantic item. m and k are the number of semantic items for T and T', respectively. $H(A_i')$ indicates the information entropy of the semantic item A_i' in trajectory T'.

In this paper, trajectory feature point selection is constrained by both spatio-temporal distance and semantic distance to avoid losing points with significant semantic information during compression. An example of a trajectory feature point is illustrated in Figure 2. Trajectory points p_1 and p_2 have the same spatio-temporal distance (i.e., *ST_Dist*), but considering semantic information, since the trajectory lacks a "School" POI, adding p_1 results in a larger change in information entropy compared to p_2. This indicates that p_1 has a larger semantic distance. When the weights of spatio-temporal distance and semantic distance are equal, the spatio-temporal semantic distance of trajectory points can be calculated using Formula (3) to determine that p_1 has a greater spatio-temporal semantic distance than p_2. This indicates that p_1 contains more spatio-temporal semantic information than p_2, leading to its selection as a trajectory feature point.

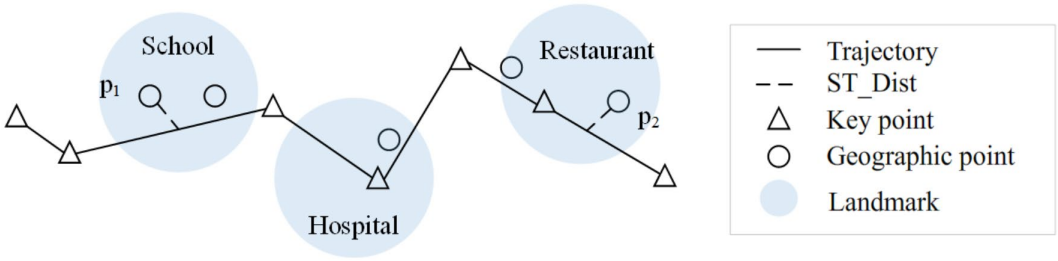

Figure 2. Example of trajectory feature points.

4. Experiments and Analysis

4.1. Datasets

The experimental data are sourced from the publicly available GPS trajectory dataset of the Geolife project collected by Microsoft Research Asia [26]. This dataset collected trajectories from 182 users over more than five years (from April 2007 to August 2012), containing 17,621 trajectories spanning 1.2 million kilometers and more than 48,000 h. Recorded by different GPS loggers and GPS phones, each trajectory consists of an ordered sequence of points, each including longitude, latitude, altitude, date, and time. The dataset captures diverse users' outdoor movements, from life routines (home and work) to entertainment and sports activities. Although this dataset is distributed across over 30 cities in China and across some cities in the USA and Europe, the majority of the data were generated in Beijing. We conducted compression experiments on a random selection of 5185 trajectories distributed in Beijing. The sampling frequency is concentrated in the range of 1~5 s, and the total number of sampling points is 734,108. The main information on trajectory points is presented in Table 1.

Table 1. Trajectory points information.

Field Name	Units	Types	Example
Longitude	Degrees	Float	39.984702
Latitude	Degrees	Float	116.318417
Altitude	Feet	Integer	492
Date	Year-month-day	String	2009-10-11
Time	Hour: minute: second	String	14:04:30

Our proposed method imposes no specific restrictions on the semantics of trajectory data. Given the ease of obtaining POI data and their frequent use in trajectory data mining, we select the POI information at the trajectory point location as the semantic for that point. To obtain the POI semantics corresponding to a trajectory point, we utilize the AMAP Open API [27] and present a semantic description of POI categories in Table 2. Figure 3 reveals the number of trajectory points with corresponding POI semantics, while Figure 4 depicts the POI semantic of the trajectory point after location matching.

Table 2. Semantic description of POI categories.

POI Category ID	Semantic Description	POI Category ID	Semantic Description
1	Catering services	8	Education services
2	Scenic spots	9	Commercial residence
3	Financial insurance services	10	Life services
4	Public facilities	11	Sports and leisure
5	Company enterprises	12	Health care services
6	Shopping services	13	Government organization
7	Transportation facilities		

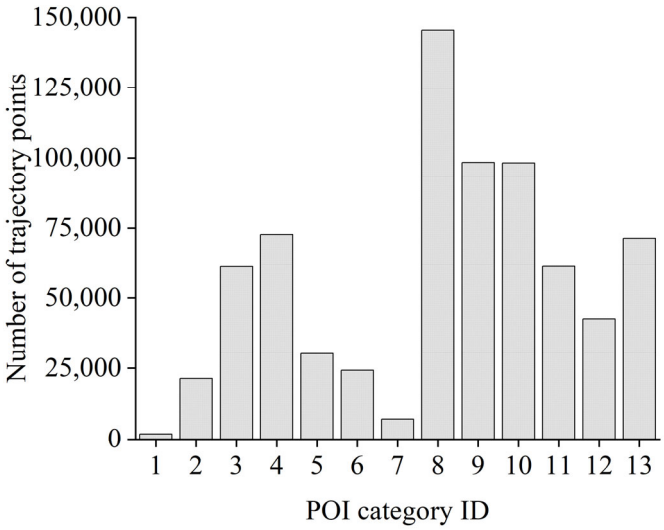

Figure 3. Number of trajectory points for POI categories.

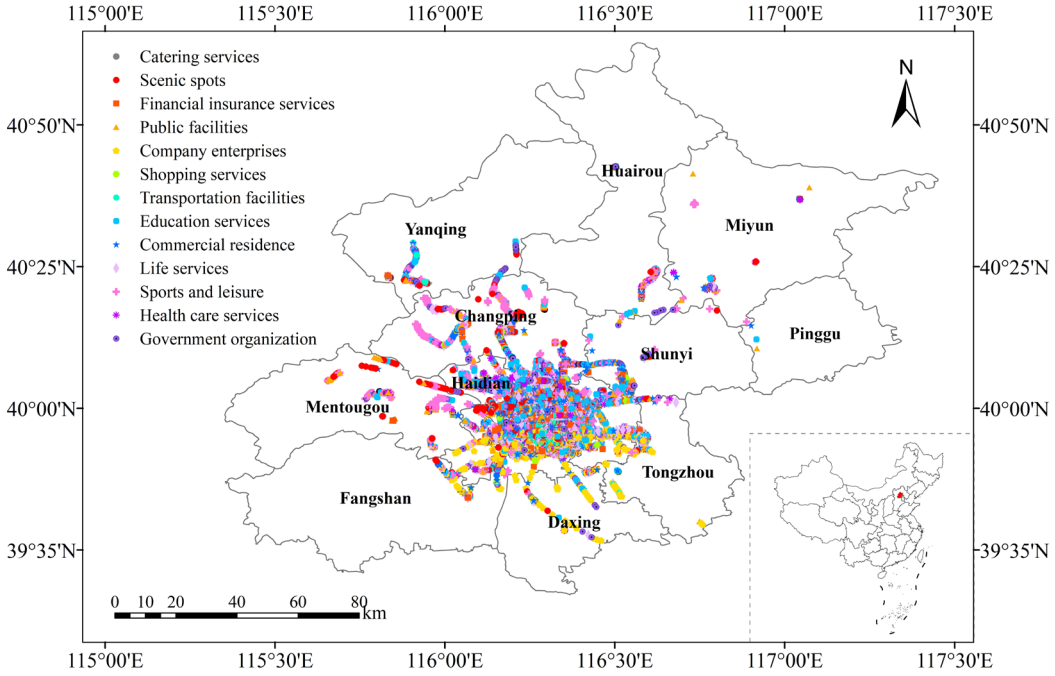

Figure 4. Trajectory points with POI semantics.

4.2. Evaluation Metrics

To comprehensively evaluate the effectiveness of the trajectory compression method in preserving important features of the original trajectory, we assess the similarity between the compressed trajectory and the original trajectory using the following three metrics:

(1) Hausdorff similarity: Hausdorff similarity is a metric for evaluating geometric similarity between the compressed and original trajectories. Hausdorff similarity [28] between trajectories T_1 and T_2 is defined as Formula (10).

$$Hausdorff_Similarity(T_1, T_2) = \frac{1}{1 + H(T_1, T_2)} \tag{10}$$

where $H(T_1, T_2)$ denotes the Hausdorff distance between T_1 and T_2;

(2) Cosine similarity: Cosine similarity is commonly used to measure similarity between vectors or data sets. In trajectory compression, cosine similarity evaluates the semantic similarity between the compressed and original trajectories, such as the types of POIs visited or semantic labels associated with the trajectory points [29]. Cosine similarity between trajectories T_1 and T_2 can be calculated with Formula (11).

$$Cosine_Similarity(T_1, T_2) = \frac{\sum_{i=1}^{n} W_{1i} * W_{2i}}{\sqrt{\sum_{i=1}^{n} (W_{1i})^2} * \sqrt{\sum_{i=1}^{n} (W_{2i})^2}} \tag{11}$$

where n is the total number of semantic categories, and W_{ji} is the corresponding weight for each semantic category;

(3) Semantic intensity: Semantic intensity is a metric of semantic–geographic combined similarity, reflecting comprehensive similarity based on semantic and geometric features [30]. It is defined by Formula (12).

$$Semantic_Intensity(T_1, T_2) = \frac{Semantic\ Similarity(T_1, T_2)}{2 - Geographic\ Similarity(T_1, T_2)} \tag{12}$$

where $Semantic\ Similarity(T_1, T_2)$ is the cosine similarity between trajectories T_1 and T_2; $Geographic\ Similarity(T_1, T_2)$ is the Hausdorff similarity between trajectories T_1 and T_2.

4.3. Experimental Results and Analysis

To analyze the proposed method's effectiveness, three trajectory compression methods are chosen for comparison:

(1) Douglas–Peucker (DP) [8]: DP is a classical spatial feature-based trajectory compression method. It uses vertical Euclidean distance from each trajectory point to the approximate trajectory as a threshold to identify trajectory feature points;
(2) Top–Down Time Ration (TD-TR) [14]: TD-TR is a spatio-temporal feature-based trajectory compression method. It considers both spatial and temporal features of trajectories and uses synchronous Euclidean distance instead of vertical Euclidean distance to identify trajectory feature points;
(3) CascadeSync [24]: CascadeSync is a semantic trajectory compression method utilizing synchronization-based clustering to represent trajectory data as multi-resolution trajectory abstractions (i.e., hierarchical ROI network). The abstracted trajectory representation preserves both the structure and semantic information of trajectories.

In experiments, the weight w_k is set to 0.5, indicating that spatial-temporal and semantic features are equally important. The compression ratio R is set to 30%, 40%, 50%, 60%, 70%, and 80%, respectively. Based on three evaluation metrics (Hausdorff similarity, cosine similarity, and semantic intensity), we conduct experiments and analysis on the proposed methods and three trajectory compression methods to evaluate compressed trajectory similarity to the original trajectory from geometric, semantic, and semantic–geographic combined perspectives. Additionally, we provided the time spent by these four methods on compressing trajectories to demonstrate the feasibility of the proposed method. All experiments described were conducted on a Dell OptiPlex 7090 computer which was produced in Xiamen, Fujian, China and is equipped with an Intel (R) Core (TM) i7-11700 CPU and Windows 11 operating system. The Python programming language was used to implement all the algorithms discussed in this paper.

4.3.1. Geometric Similarity Evaluation

Figure 5 displays the results of geometric similarity evaluation using the Hausdorff similarity metric between compressed trajectories obtained by four compression methods and original trajectories.

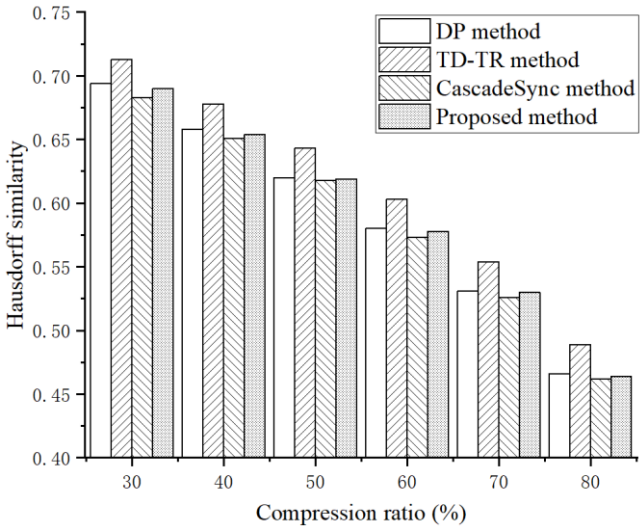

Figure 5. Hausdorff similarity evaluation results.

Results show decreasing Hausdorff similarity as the compression ratio increases, indicating greater loss of geometric features in compressed trajectories at higher compression ratios, leading to lower geometric similarity with original trajectories. Under the same compression ratio, DP and TD-TR outperform the proposed method and CascadeSync in Hausdorff similarity. This is due to DP and TD-TR primarily focusing on compressing trajectories considering geometric features, resulting in better geometric similarity between compressed and original trajectories. Furthermore, TD-TR incorporates temporal features and utilizes synchronous Euclidean distance instead of vertical Euclidean distance in DP, resulting in superior geometric similarity results compared to the other three methods.

As shown in Table 3, CascadeSync and the proposed method exhibit slightly lower Hausdorff similarity than DP and TD-TR. For example, when the compression ratio is 40%, the Hausdorff similarities of the proposed method, TD-TR, DP, and CascadeSync are 0.658, 0.678, 0.654, and 0.651, respectively. This is because CascadeSync and our proposed method consider semantic features of trajectories, potentially trading off geometric features to preserve significant semantic points. However, the geometric similarity results of all four methods are very close. For example, when the compression ratio is 50%, Hausdorff similarities of CascadeSync and the proposed method are 0.618 and 0.619, and those of DP and TD-TR are 0.620 and 0.643, respectively. This indicates that the four methods can maintain the geometric features of the original trajectory well.

Table 3. Hausdorff similarities of four methods.

Compression Ratio	Proposed Method	TD-TR Method	DP Method	CascadeSync Method
30%	0.694	0.713	0.690	0.683
40%	0.658	0.678	0.654	0.651
50%	0.620	0.643	0.619	0.618
60%	0.580	0.603	0.578	0.573
70%	0.531	0.554	0.530	0.526
80%	0.466	0.489	0.464	0.462

4.3.2. Semantic Similarity Evaluation

Figure 6 displays the semantic similarity evaluation results, showing the cosine similarity between compressed trajectories from four methods and original trajectories.

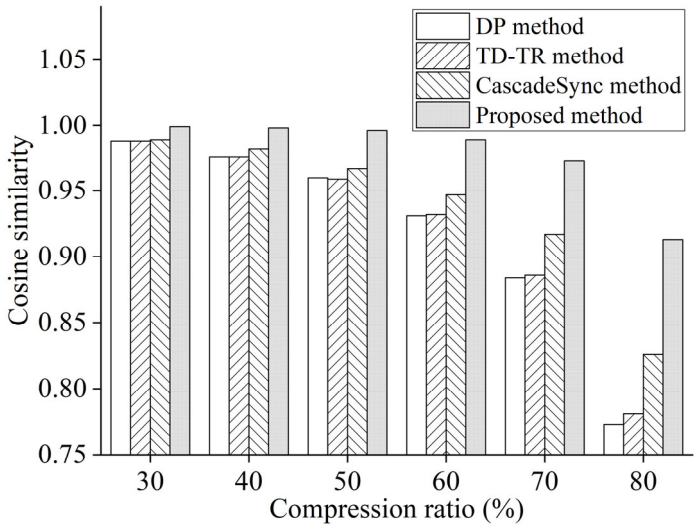

Figure 6. Cosine similarity evaluation results.

The results demonstrate decreasing cosine similarity with increasing compression ratio, indicating greater loss of semantic features and lower semantic similarity at higher

compression ratios. Under the same compression ratio, our proposed method significantly outperforms DP, TD-TR, and CascadeSync, with a more obvious advantage as the compression ratio increases. This indicates that the proposed method better maintains semantic features between compressed and original trajectories.

As shown in Table 4, the proposed method exhibits higher cosine similarity compared to DP, TD-TR, and CascadeSync methods under the same compression ratio. Moreover, while the cosine similarity of DP, TD-TR, and CascadeSync decreases obviously as the compression ratio increases; the cosine similarity of our proposed method remains relatively higher, exceeding 0.9. For example, when the compression ratio is 30%, the cosine similarity of DP, TD-TR, our proposed method, and CascadeSync are 0.988, 0.988, 0.999, and 0.989, respectively. And under a compression ratio of 80%, the values of these four methods are 0.773, 0.781, 0.913, and 0.826, respectively. This is due to the spatio-temporal semantic distance model adopted in feature point selection in the proposed method. In addition, unlike CascadeSync, which fixes semantic feature points based on a priori knowledge, our proposed method introduces information entropy as the semantic feature metric, so measurement results are not influenced by prior knowledge but are closer to the trajectory itself. Therefore, compared with DP, TD-TR, and CascadeSync, our proposed method obtains better semantic similarity evaluation results.

Table 4. Cosine similarity of four methods.

Compression Ratio	Proposed Method	TD-TR Method	DP Method	CascadeSync Method
30%	0.999	0.988	0.988	0.989
40%	0.998	0.976	0.976	0.982
50%	0.996	0.959	0.960	0.967
60%	0.989	0.932	0.931	0.947
70%	0.973	0.886	0.884	0.917
80%	0.913	0.781	0.773	0.826

To show differences in feature point retention among the four methods, we randomly select a trajectory and visualize trajectory compression results through Figure 7. Since a single trajectory contains a large number of trajectory points, we set the compression ratio to 80% to clearly show differences in trajectory compression results of the four methods in terms of retaining feature points. Dots of different colors represent trajectory points with different types of POI, and semantic loss of compressed trajectory can be judged by comparing the color change in trajectory points before and after trajectory compression. To observe the difference in semantic features of the four methods, we use black rectangle boxes to mark areas of trajectory points with obvious changes in semantic features in Figure 7.

As shown in Figure 7, the original trajectory has eight POI categories, and our proposed method retains all of them in the compressed trajectory. In contrast, the compression results of other methods, DP, TD-TR, and CascadeSync, exhibit varying degrees of POI category loss. The comparison areas marked by black rectangular boxes (box 1 and box 2) illustrate our method's significant advantage in retaining POI variety. Taking box 1 as an example, our proposed method retains four POI categories, while other methods retain only two. In box 2, our proposed method retains both two POI categories, whereas other methods retain only one. Overall, in terms of geometric features, the compressed trajectories from our proposed method, DP and TD-TR, all largely retain the main geometric features of the original trajectory. However, the CascadeSync trajectory points are slightly shifted due to the synchronous clustering method used to synchronize trajectory points at different locations to the same position through interaction. In terms of semantic features, our proposed method better retains original semantic information compared to DP, TD-TR, and CascadeSync methods, making the compressed trajectories more interpretable and better supporting trajectory data mining and deeper analysis.

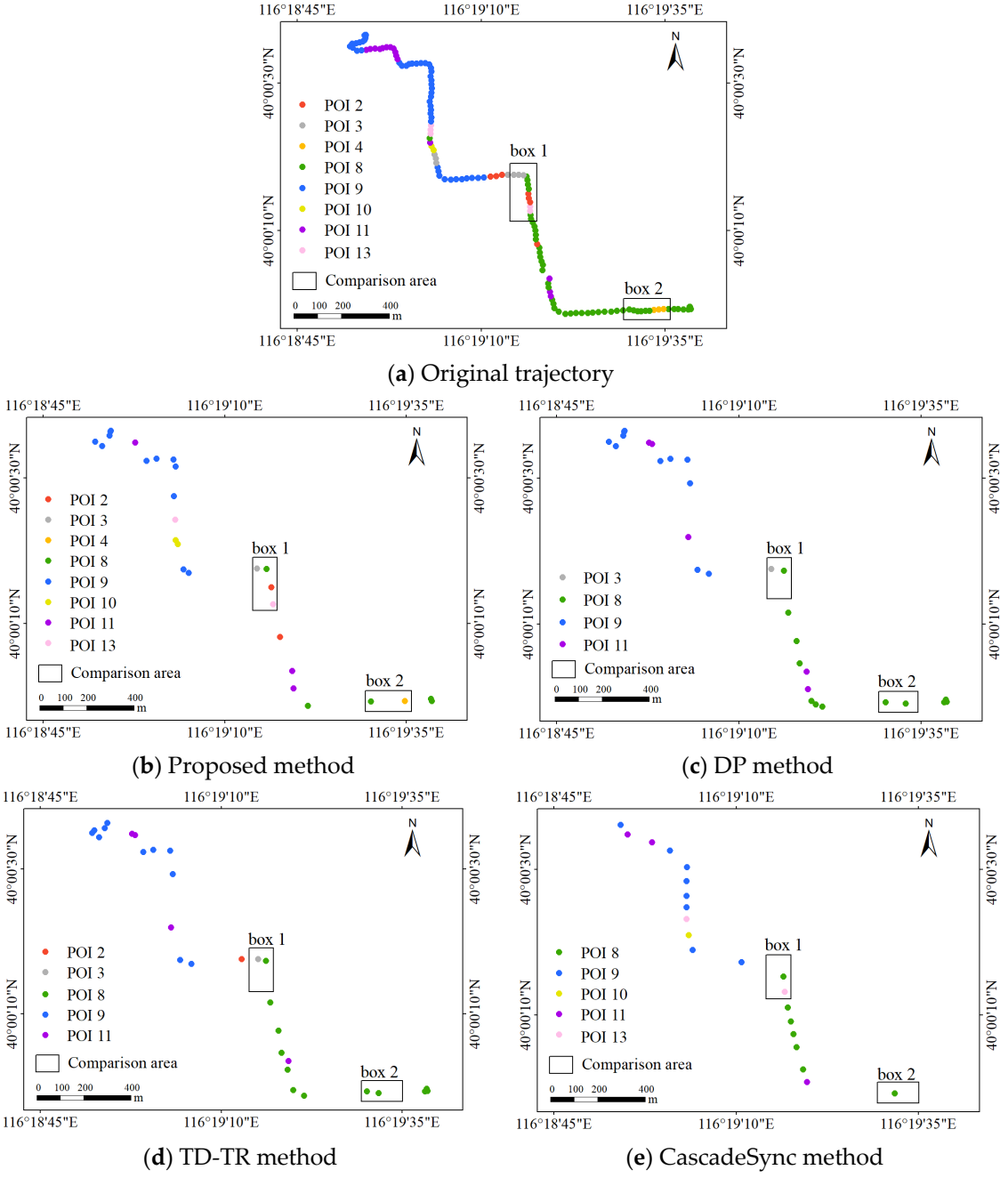

Figure 7. Trajectory compression results of four methods. Points with different colors represent different semantic categories of POI. The comparison areas marked by black rectangular boxes, box 1 and box 2, are the areas with significant POI semantic changes.

4.3.3. Semantic–Geographic Combined Similarity Evaluation

Figure 8 shows the semantic intensity evaluation results of compressed and original trajectories from four compression methods. It shows that the semantic intensity decreases as the compression rate increases for all methods, indicating greater loss of spatio-temporal semantic features and worse overall similarity at higher compression. As shown in Figure 8, under the same compression ratio, the semantic intensity of the DP method is the lowest. TD-TR and CascadeSync exhibit nearly identical semantic intensities. As the compression ratio increases, the semantic intensity of CascadeSync is slightly higher than TD-TR. The reason for this is that DP only considers trajectory spatial features, while TD-TR also considers temporal information. Therefore, TD-TR performs better than DP in terms of spatio-temporal similarity. In addition, compared to DP and TD-TR, CascadeSync exhibits

slightly higher semantic intensity but is still lower than that of our proposed method. That is because CascadeSync combines semantic fixed regions to synchronize trajectory points with similar locations by clustering and considers both semantic and spatial features of trajectory data. In other words, CascadeSync can exhibit higher semantic similarity than DP and TD-TR methods.

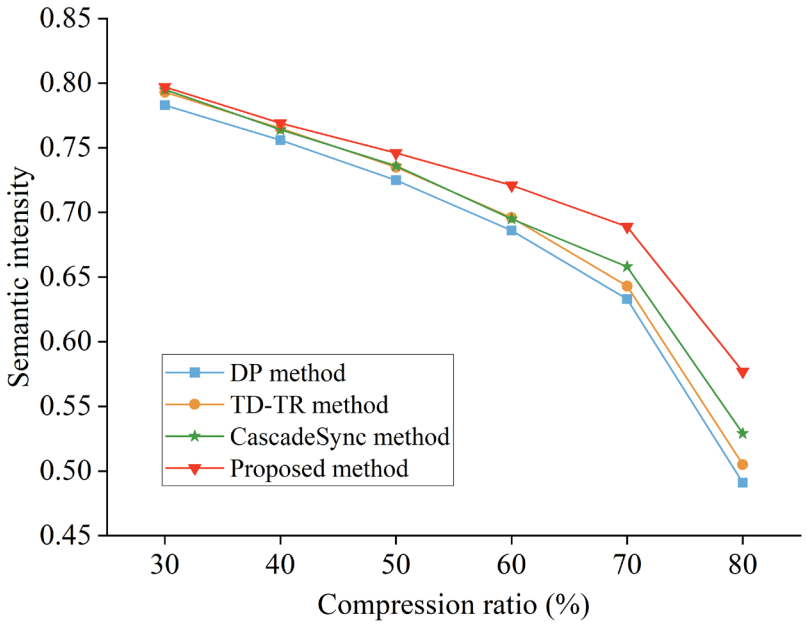

Figure 8. Semantic intensity evaluation results.

As shown in Figure 8, in terms of semantic intensity evaluation, the proposed method significantly outperforms the DP, TD-TR, and CascadeSync methods. This is due to the fact that, when finding trajectory feature points, the proposed method not only considers the spatio-temporal morphological similarity of the compressed trajectories but also introduces information entropy as the semantic similarity metric factor. This ensures that the spatio-temporal semantic similarities between compressed and original trajectories are maintained to the greatest extent possible. The specific values of the semantic intensity evaluation results of the four methods are given in Table 5. The results show that, in terms of overall geographic–semantic similarity, the proposed method can consider both the spatio-temporal semantic characteristics of the trajectories by constraining the selection of trajectory feature points through the spatio-temporal semantic distance, making the overall similarity between the compressed trajectories and the original trajectories better.

Table 5. Semantic intensity evaluation of four methods.

Compression Ratio	Proposed Method	TD-TR Method	DP Method	CascadeSync Method
30%	0.797	0.793	0.783	0.795
40%	0.769	0.765	0.756	0.764
50%	0.746	0.735	0.725	0.736
60%	0.721	0.696	0.686	0.695
70%	0.689	0.643	0.633	0.658
80%	0.577	0.505	0.491	0.529

4.3.4. Execution Time Evaluation

We recorded the time taken by the four methods to process a single trajectory during compression of 5185 trajectories, each with an average of 141.58 points. Figure 9 shows the execution time evaluation results of the four compression methods. The horizontal axis represents the compression ratio (%), while the vertical axis represents execution time

(seconds). It illustrates the average time required to compress a single trajectory for each of the four methods at compression ratios of 30%, 40%, 50%, 60%, 70%, and 80%.

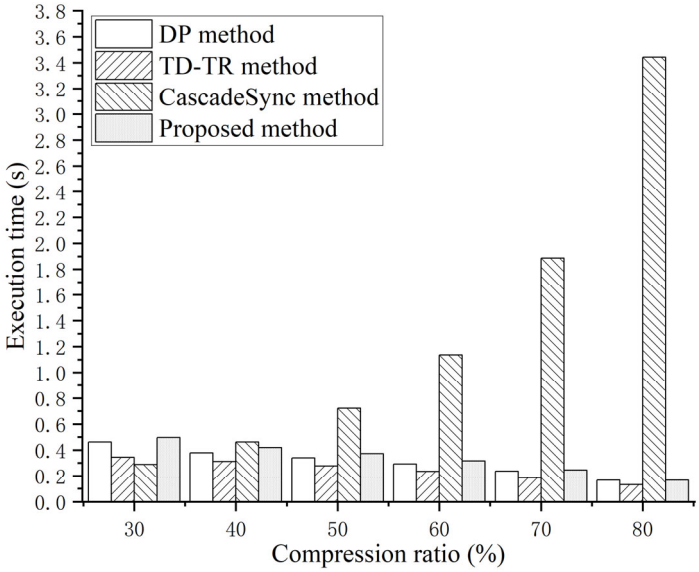

Figure 9. Execution time evaluation results.

It can be observed that as the compression ratio increases, the execution time for the proposed method, TD-TR, and DP shows a decreasing trend, while the CascadeSync shows an increasing trend. The first three methods use a scheme where they start with an empty compressed trajectory and select points from the original trajectory into the compressed trajectory until the target number of points is reached. The higher the compression ratio, the fewer points the algorithm needs to select and, thus, the shorter the execution time. In contrast, the CascadeSync employs a clustering approach during compression. This method starts with the original trajectory and continuously aggregates points that are close to new single points until the compressed trajectory has the target number of points. The higher the compression ratio, the more clustering operations the algorithm needs to perform, resulting in longer execution times.

In terms of time consumption, TD-TR performs best; the proposed method is slightly slower than TD-TR and DP, but all three are very close overall. The differences in time consumption of these three methods are mainly due to the distance measurement approaches each algorithm employs. The proposed method uses a distance measurement approach that considers more factors, making it relatively more complex and, thus, resulting in higher time consumption. When the compression ratio exceeds 50%, the time consumption of the CascadeSync method is significantly higher than that of the other three methods and increases sharply with the compression ratio. This is due to the more complex clustering approach employed by CascadeSync, which involves more iterations as the compression ratio increases.

In summary, the proposed method achieves similar results in terms of time consumption compared to the TD-TR and DP while obtaining the best geometric similarity and semantic similarity evaluations.

4.3.5. Parameter Analysis

To investigate the effect of different weight values w_k in our proposed method on the similarity between the compressed and original trajectories, we conducted geometric similarity, semantic similarity, and spatio-temporal semantic analysis experiments with w_k varying in steps of 0.1 over the range [0, 1]. The results are shown in Figures 9–11.

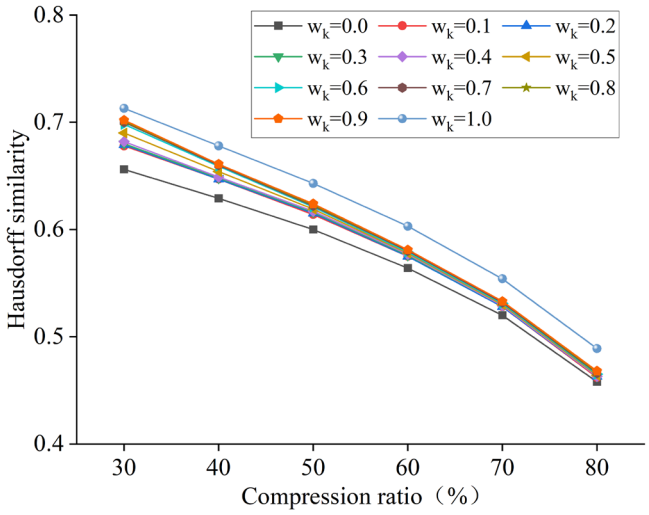

Figure 10. Geometric similarity for different w_k.

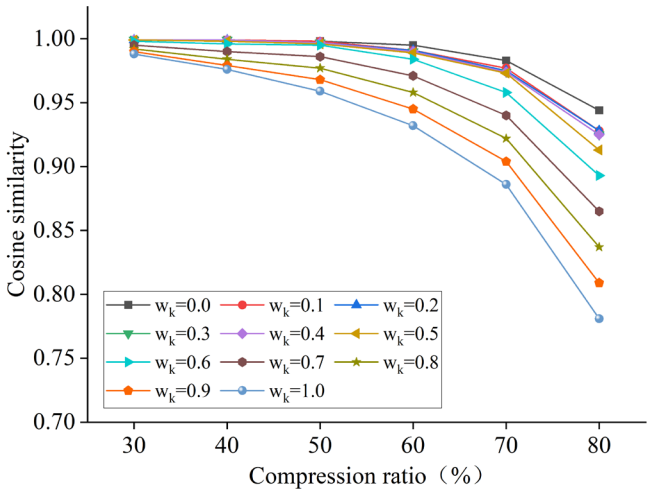

Figure 11. Semantic similarity for different w_k.

Figure 10 shows the result of geometric similarity analysis between compressed and original trajectories for different w_k. It shows that geometric similarity is negatively correlated with w_k at the same compression ratio. When $w_k = 0$, the Hausdorff similarity between compressed and original trajectories is the lowest; as the method's spatio-temporal distance weight is zero, it only considers semantic features and ignores the spatio-temporal features when selecting trajectory feature points, so the geometric similarity is the lowest. When $w_k \geq 0$, the Hausdorff similarity between compressed and original trajectories slowly rises with increasing w_k. With the incorporation of spatio-temporal features, geometric similarity substantially improves, indicating the importance of spatio-temporal features. The Hausdorff similarity reaches the maximum when $w_k = 1$, as the proposed method then only considers spatio-temporal features when selecting trajectory feature points.

Figure 11 shows the semantic similarity evaluation results between compressed and original trajectories for different w_k. The result shows that the cosine similarity is negatively correlated with w_k at the same compression ratio. The cosine similarity is the lowest when $w_k = 1$, due to this method, only considering spatio-temporal features of the trajectory, ignoring semantic features. As w_k decreases, semantic features become more important in feature point selection, so the cosine similarity between compressed and original trajectories increases accordingly. The cosine similarity reaches a maximum value when $w_k = 0$ because the proposed method only considers semantic features of the trajectory during trajectory feature point selection at this time.

Figure 12 shows the semantic intensity evaluation results between compressed and original trajectories for different w_k. It can be seen that semantic intensity decreases with increasing compression ratio. Under the same compression ratio, semantic intensity is relatively optimal when $w_k = 0.5$. This indicates that balancing the relationship between spatio-temporal and semantic features is important to maintain spatio-temporal semantic similarity between the compressed and original trajectories. The weights of spatio-temporal and semantic features should be relatively balanced in feature point selection to maximize spatio-temporal semantic similarity between the compressed and original trajectories.

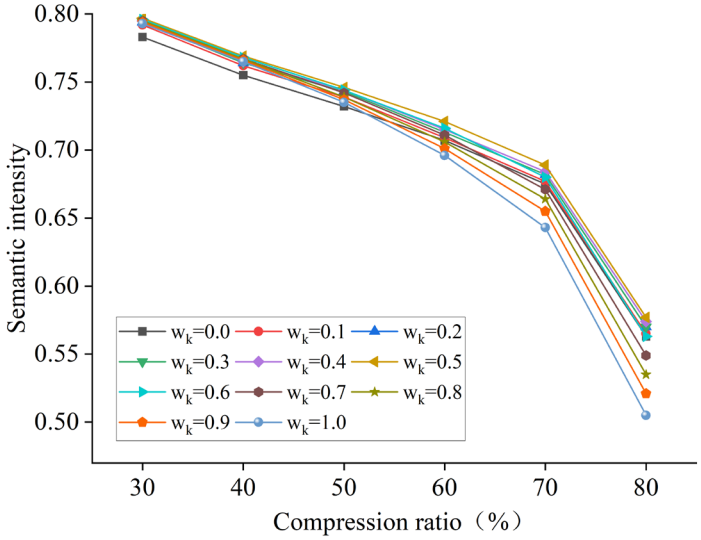

Figure 12. Semantic intensity for different w_k.

5. Conclusions

This paper proposes a spatio-temporal semantic constrained trajectory data compression method, introducing information entropy to measure semantic features of trajectories, utilizing synchronous Euclidean distance to measure spatio-temporal features and fusing these to construct a spatio-temporal semantic constrained distance model. Based on the principle of maximum spatio-temporal semantic distance, trajectory feature points are selected to maximize spatio-temporal semantic similarity between compressed and original trajectories. The experimental results indicate that the proposed method maintains the ability to preserve the geometric features of trajectory data on par with existing algorithms while significantly retaining more extensive semantic information. This enhances the explainability of the compressed trajectory to facilitate large-scale trajectory data mining and spatio-temporal semantic analysis. However, the proposed method in this paper is an offline compression algorithm requiring loading the entire trajectory at once for compression, making it unsuitable for real-time trajectory data compression. In the future, we will consider designing an algorithm capable of online trajectory data compression and investigate the feasibility of parallelizing this algorithm to improve compression speed. We aim to expand this research in the future to meet varying demands of both offline and online trajectory data compression.

Author Contributions: Conceptualization, Y.Z. (Yan Zhou); formal analysis, Y.Z. (Yan Zhou); funding acquisition, Y.Z. (Yan Zhou); investigation, Y.Z. (Yunhan Zhang) and F.Z.; methodology, Y.Z. (Yan Zhou) and Y.Z. (Yunhan Zhang); supervision, Y.Z. (Yan Zhou); validation, Y.Z. (Yunhan Zhang); writing—original draft, Y.Z. (Yunhan Zhang); writing—review and editing, Y.Z. (Yan Zhou), Y.Z. (Yeting Zhang), and X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 41871321), the National Key Research and Development Program of China (2022YFC3005702), and

the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources (KF-2021-06-033).

Data Availability Statement: The datasets analyzed in this study are available on the website https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/ (accessed on 24 June 2023).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Qinghua, W. Online GPS Trajectory Data Compression Algorithm Based on Relative Synchronous Euclidean Distance Filtering. *Comput. Appl. Softw.* **2018**, *35*, 282–288.
2. Chen, H.B.; Chen, X. A Trajectory Ensemble-Compression Algorithm Based on Finite Element Method. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 334. [CrossRef]
3. Sun, S.; Chen, Y.; Piao, Z.J.; Zhang, J.S. Vessel AIS Trajectory Online Compression Based on Scan-Pick-Move Algorithm Added Sliding Window. *IEEE Access* **2020**, *8*, 109350–109359. [CrossRef]
4. Ji, Y.Y.; Qi, L.; Balling, R. A Dynamic Adaptive Grating Algorithm for Ais-Based Ship Trajectory Compression. *J. Navig.* **2022**, *75*, 213–229. [CrossRef]
5. Zhang, D.X.; Chang, Z.H.; Wu, S.; Yuan, Y.; Tan, K.L.; Chen, G. Continuous Trajectory Similarity Search for Online Outlier Detection. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 4690–4704. [CrossRef]
6. Zhong, Y.L.; Kong, J.L.; Zhang, J.Q.; Jiang, Y.Z.; Fan, X.; Wang, Z.Y. A Trajectory Data Compression Algorithm Based on Spatio-Temporal Characteristics. *PeerJ Comput. Sci.* **2022**, *8*, 1112–1135. [CrossRef] [PubMed]
7. Richard, B. On The Approximation of Curves by Line Segments Using Dynamic Programming. *Arch. Intern. Med.* **1961**, *4*, 284.
8. Douglas, D.H.; Peucker, T.K. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Lineor Its Caricature. *Cartographica* **1973**, *10*, 112–122. [CrossRef]
9. Hershberger, J.; Snoeyink, J. Speeding Up the Douglas-Peucker Line-Simplification Algorithm. In Proceedings of the International Symposium on Spatial Data Handling, Beijing, China, 3–6 November 2000; pp. 134–143.
10. Liu, J.X.; Li, H.H.; Yang, Z.L.; Wu, K.F.; Liu, Y.; Liu, R.W. Adaptive Douglas-Peucker Algorithm with Automatic Thresholding for AIS-Based Vessel Trajectory Compression. *IEEE Access* **2019**, *7*, 150677–150692. [CrossRef]
11. Tang, C.H.; Wang, H.; Zhao, J.H.; Tang, Y.Q.; Yan, H.R.; Xiao, Y.J. A Method for Compressing AIS Trajectory Data Based on the Adaptive-Threshold Douglas-Peucker Algorithm. *Ocean. Eng.* **2021**, *232*, 109041–109054. [CrossRef]
12. Huang, Y.; Li, Y.; Zhang, Z.F.; Liu, R.W. GPU-Accelerated Compression and Visualization of Large-Scale Vessel Trajectories in Maritime IoT Industries. *IEEE Internet Things J.* **2020**, *7*, 10794–10812. [CrossRef]
13. Li, Y.T.; Sun, W.W. Trajectory Compression Algorithm Based on Recurrent Neural Network. *Comput. Sci.* **2020**, *47*, 102–107.
14. Meratnia, N.; By, R.D. Spatiotemporal Compression Techniques for Moving Point Objects. *Adv. Database Technol.* **2004**, *2992*, 765–782.
15. Zhu, F.; Ma, Z. Ship Trajectory Online Compression Algorithm Considering Handling Patterns. *IEEE Access* **2021**, *9*, 70182–70191. [CrossRef]
16. Potamias, M.; Patroumpas, K.; Sellis, T. Sampling Trajectory Streams with Spatiotemporal Criteria. In Proceedings of the 18th International Conference on Scientific and Statistical Database Management (SSDBM'06), Vienna, Austria, 3–5 July 2006; pp. 275–284.
17. Muckell, J.; Olsen, P.W.; Hwang, J.H.; Lawson, C.T.; Ravi, S.S. Compression of Trajectory Data: A Comprehensive Evaluation and New Approach. *Geoinformatica* **2014**, *18*, 435–460. [CrossRef]
18. Liu, S.J.; Chen, G.; Wei, L.; Li, G.Q. A Novel Compression Approach for Truck GPS Trajectory Data. *IET Intell. Transp. Syst.* **2021**, *15*, 74–83. [CrossRef]
19. Zhao, P.; Zhao, Q.P.; Zhang, C.X.; Su, G.; Zhang, Q.; Rao, W.X. CLEAN: Frequent Pattern-Based Trajectory Compression and Computation on Road Networks. *China Commun.* **2020**, *17*, 119–136. [CrossRef]
20. Zheng, K.; Zhao, Y.; Lian, D.F.; Zheng, B.L.; Liu, G.F.; Zhou, X.F. Reference-Based Framework for Spatio-Temporal Trajectory Compression and Query Processing. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 2227–2240. [CrossRef]
21. Richter, K.F.; Schmid, F.; Laube, P. Semantic Trajectory Compression: Representing Urban Movement in a Nutshell. *J. Spat. Inf. Sci.* **2012**, 3–30. [CrossRef]
22. Mirvahabi, S.; Ali Abbaspour, R.; Claramunt, C. A Flexible Trajectory Compression Algorithm for Multi-Modal Transportation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *10*, 501–508. [CrossRef]
23. Zhang, K.X.; Zhao, D.B.; Liu, W.K. Online Vehicle Trajectory Compression Algorithm Based on Motion Pattern Recognition. *IET Intell. Transp. Syst.* **2022**, *16*, 998–1010. [CrossRef]
24. Gao, C.M.; Zhao, Y.; Wu, R.Z.; Yang, Q.L.; Shao, J.M. Semantic Trajectory Compression Via Multi-Resolution Synchronization-Based Clustering. *Knowl. -Based Syst.* **2019**, *174*, 177–193. [CrossRef]
25. Kharazmi, O.; Balakrishnan, N.; Ozonur, D. Jensen-Discrete Information Generating Function with an Application to Image Processing. *Soft Comput.* **2023**, *27*, 4543–4552. [CrossRef]

26. Geolife GPS Trajectory Dataset User Guide. Available online: https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/ (accessed on 24 June 2023).
27. AMAP Open API. Available online: https://lbs.amap.com/api/ios-sdk/guide/map-data/poi/ (accessed on 24 April 2023).
28. Atallah, M.J. A Linear Time Algorithm for The Hausdorff Distance Between Convex Polygons. *Inf. Process. Lett.* **1983**, *17*, 207–209. [CrossRef]
29. Lv, M.; Chen, L.; Chen, G. Mining user similarity based on routine activities. *Inf. Sci.* **2013**, *236*, 17–32. [CrossRef]
30. Wan, Y.; Zhou, C.H.; Pei, T. Semantic-Geographic Trajectory Pattern Mining Based on a New Similarity Measurement. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 212. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.